

z/OS
2.5

*IBM z/OS Management Facility
Programming Guide*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 1307](#).

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-06-26

© **Copyright International Business Machines Corporation 2013, 2023.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures.....	ix
Tables.....	xxxiii
About this document.....	lix
Who should use this document.....	lix
Where to find more information.....	lix
How to send your comments to IBM.....	lxi
If you have a technical problem.....	lxi
Summary of changes.....	lxiii
Summary of changes for z/OSMF Programming Guide for Version 2 Release 5 (V2R5).....	lxiii
Summary of changes for z/OSMF Programming Guide for Version 2 Release 4 (V2R4).....	lxv
Information applicable to all releases.....	lxx
Chapter 1. Using the z/OSMF REST services.....	1
Application Linking Manager interface services.....	8
Event types, requestors, and handlers shipped with z/OSMF.....	10
Register an event type.....	24
Register an event handler.....	26
Obtain a list of all tasks that are eligible to be handlers.....	29
Obtain a list of handlers for an event type.....	30
Unregister an event handler.....	31
Unregister an event type.....	32
Application server routing services.....	33
Retrieve data from an application server.....	36
Update data for an application server.....	40
Delete data from an application server.....	43
Cloud provisioning services.....	46
Resource pool services.....	50
Resource management services.....	102
Software services template services.....	196
Published software service template services.....	283
Software services instance services.....	319
Software service instance name services.....	390
Data persistence services.....	400
Persist user or application data.....	401
Retrieve persisted user or application data.....	403
Delete persisted user or application data.....	406
Multisystem routing services.....	408
Retrieve data from one or more systems.....	413
Update data for one or more systems.....	420
Delete data from one or more systems.....	426
Authenticate with a secondary z/OSMF instance.....	432
Authenticate with an HTTP proxy server.....	434
MVS subsystem services.....	435
List MVS subsystems.....	436
Notification services.....	438

Get all of the notifications received by the current user.....	439
Send a notification from a z/OSMF task, when the content is the message from the bundle file ..	441
Send a notification and mail from a z/OSMF task or z/OSMF user.....	443
Security Configuration Assistant services.....	446
Validate security resources defined in an HTTP request body.....	447
Validate security resources defined in a security descriptor file.....	451
Provision security resources defined in an HTTP request body.....	455
Software management services.....	458
List the software instances defined to z/OSMF.....	461
Retrieve the properties of a software instance.....	464
List the data sets included in a software instance.....	470
Add a new software instance.....	476
Export a defined software instance.....	482
Modify the properties of a software instance.....	488
Load the products, features, and FMIDs for a software instance.....	495
Delete a software instance.....	501
Deleting the Temporary Catalog Aliases.....	502
List the portable software instances defined to z/OSMF.....	507
Deploy a software instance.....	509
Delete a Deployment.....	519
Retrieve the properties of a portable software instance.....	520
Add a new portable software instance.....	525
Delete a portable software instance.....	530
Missing Critical Updates.....	531
Missing FIXCAT Updates.....	536
Software Update Search.....	540
Retrieve the z/OS system UUID.....	544
Storage management services.....	547
Get a list of storage groups.....	549
Get a storage group definition.....	560
Get a list of volumes.....	571
Get a volume definition.....	576
Get a list of data classes.....	580
Get a data class definition.....	597
Get a list of storage classes.....	614
Get a storage class definition.....	621
Define a volume list to a storage group.....	628
Validate an SCDS.....	634
Activate an SCDS.....	636
Get an SCDS activation result.....	638
Error reporting categories.....	639
Sysplex management services.....	658
Get a list of CFRM policies.....	659
Get CFRM policy properties.....	661
Activate CFRM policy.....	667
Rename existing CFRM policy.....	668
Modify CFRM policy properties.....	670
Add CFRM policies to administrative policy.....	676
Copy existing CFRM policy with a new name.....	684
Delete multiple CFRM policies from an administrative policy.....	685
Error report.....	686
Topology services.....	687
List the systems defined to z/OSMF.....	689
List the groups defined to z/OSMF.....	692
List the systems included in a group.....	694
List the sysplexes defined to z/OSMF.....	697
List the systems included in a sysplex.....	698
List the systems included in a CPC.....	701

Validate the status of the selected system.....	703
Check all MVS systems in the local plex.....	709
TSO/E address space services.....	714
Start or reconnect to a TSO/E address space.....	718
Start an application in a TSO/E address space.....	722
Issue a TSO/E command with z/OSMF REST API.....	725
Get the response to a command that was issued with the TSO/E REST API.....	734
Send messages to a TSO/E address space.....	737
Send messages to an application.....	739
Ping a TSO/E address space.....	740
Receive messages from a TSO/E address space.....	742
Receive messages from an application.....	743
End a TSO/E address space.....	744
WLM resource pooling services.....	746
Prime a WLM resource pool.....	747
Delete a WLM resource pool.....	749
Construct a WLM service definition.....	751
Construct a WLM service definition with remove and install.....	753
RMF metering services.....	754
Get metered data.....	755
Get DDS server time data.....	758
z/OS Compliance REST Interface.....	760
POST the request for compliance facts	762
z/OS console services.....	763
Issue a command from a system console.....	765
Get a command response.....	782
Get the detect result for unsolicited messages.....	788
Get messages from a hardcopy log.....	795
z/OS system variable services.....	802
Create or update system variables.....	803
Get system variables.....	805
Import system variables.....	809
Export system variables.....	810
Delete system variables.....	812
z/OS data set and file REST interface.....	813
List the z/OS data sets on a system.....	820
List the members of a z/OS data set.....	823
Retrieve the contents of a z/OS data set or member.....	826
Write data to a z/OS data set or member.....	831
Create a sequential or partitioned data set.....	837
Delete a sequential and partitioned data set.....	840
Delete a partitioned data set member.....	842
z/OS data set and member utilities.....	844
Access Method Services Interface.....	848
List the files and directories of a UNIX file path.....	850
Retrieve the contents of a z/OS UNIX file.....	855
Write data to a z/OS UNIX file.....	859
Create a UNIX file or directory.....	862
Delete a UNIX file or directory.....	865
z/OS UNIX file utilities.....	867
List z/OS UNIX Filesystems.....	874
Create a z/OS UNIX zFS filesystem.....	876
Delete z/OS UNIX zFS Filesystem.....	878
Mount a z/OS UNIX file system.....	879
Unmount a UNIX file system.....	881
JSON document specifications for z/OS data set and file REST interface requests.....	883
Error reporting categories.....	893
z/OS jobs REST interface.....	900

Obtain the status of a job.....	905
List the jobs for an owner, prefix, or job ID.....	909
List the spool files for a job.....	912
Retrieve the contents of a job spool file.....	914
Submit a job.....	918
Hold a job.....	923
Release a job.....	926
Change the job class.....	928
Cancel a job.....	931
Cancel a job and purge its output.....	933
JSON document specifications for z/OS jobs REST interface requests.....	936
Error reporting categories.....	943
z/OS Management Services Catalog services.....	953
List the services in the catalog.....	956
Get the details of a service from the catalog.....	962
Get the list of categories.....	975
Get category details.....	978
List the service submissions.....	979
Create a new service submission.....	982
Delete a service submission.....	989
Get the details of a service submission.....	990
Modify an existing service submission.....	995
Perform an action on a service submission.....	997
Get the list of JCL JOB statements.....	999
Get the list of target systems.....	1001
z/OSMF information retrieval service.....	1002
Retrieve z/OSMF information.....	1003
z/OSMF settings services.....	1006
Persist z/OSMF settings for a user or plug-in.....	1008
Retrieve z/OSMF settings data.....	1009
Delete z/OSMF settings data.....	1011
z/OSMF authentication services.....	1013
Log in to the z/OSMF server.....	1014
Change the user password or passphrase.....	1016
Log out of the z/OSMF server.....	1019
z/OSMF workflow services.....	1020
Create a workflow.....	1023
Get the properties of a workflow.....	1031
List the workflows for a system or sysplex.....	1054
Start a workflow.....	1056
Cancel a workflow.....	1061
Delete a workflow.....	1063
Retrieve a workflow definition.....	1065
Archive a workflow instance.....	1078
List the archived workflows for a system.....	1080
Get the properties of an archived workflow.....	1082
Delete an archived workflow.....	1096
Chapter 2. Creating workflow definitions for z/OS.....	1099
Terms you should know.....	1100
The Workflows task schema.....	1102
Creating the workflow definition file.....	1102
Structure of a workflow definition file.....	1103
Creating and viewing the workflow definition file.....	1104
Sample XML files for your reference.....	1105
References to external files.....	1106
Defining entities for a workflow.....	1107

Specifying the workflow root element.....	1108
Specifying the workflow metadata.....	1109
Including a manifest of translated text.....	1112
Enabling a workflow definition file for future upgrades.....	1113
Collecting user feedback.....	1115
Defining steps for your workflow.....	1117
Parent steps and leaf steps.....	1117
Template steps.....	1119
REST steps.....	1132
Calling steps.....	1137
Automated steps.....	1143
Making a step conditional.....	1147
runAsUser identity for a step.....	1149
Using translatable strings.....	1154
Using rich translatable strings.....	1154
Defining variables for your workflow.....	1154
Using Velocity templates for variable substitution and other functions.....	1155
Specifying the variable element and its attributes.....	1156
Sub-elements of the variable element.....	1158
Using the element atCreate to qualify a variable definition	1160
How to refer to a variable.....	1161
Workflow internal variables	1163
System variables.....	1165
Array variables.....	1168
Caller scope variables.....	1169
Providing a workflow variable input file.....	1170
Workflow XML reference.....	1173
Chapter 3. Creating your own z/OSMF plug-ins.....	1223
Process of joining the z/OSMF ecosystem.....	1225
Example of a third party z/OSMF plug-in.....	1226
Developing web-based applications.....	1226
Using the z/OSMF core JavaScript APIs.....	1227
Using the Application Linking Manager JavaScript APIs.....	1240
Logging client messages in the z/OSMF log.....	1251
Retrieving files and resources for your application.....	1256
Authoring end user assistance.....	1257
Creating help plug-ins.....	1258
Adding links to help plug-ins.....	1267
Adding your applications to z/OSMF.....	1268
Securing your applications.....	1274
Chapter 4. Preparing software to exploit cloud provisioning.....	1277
Appendix A. Enabling tracing for the z/OS jobs REST interface.....	1283
Appendix B. Creating product information files for the Software Management task.....	1285
Appendix C. Understanding the Portable Software Instance descriptor file.....	1289
Appendix D. Software Management workflow variables.....	1297
Appendix E. Accessibility.....	1305
Notices.....	1307

Terms and conditions for product documentation.....	1308
IBM Online Privacy Statement.....	1309
Policy for unsupported hardware.....	1309
Minimum supported hardware.....	1309
Trademarks.....	1310
Index.....	1311

Figures

1. Key components in the application linking process.....	8
2. Registering an event type: request content.....	24
3. Example: Registering an event type.....	25
4. Example: Returned results of a successful event registration.....	26
5. Example: Returned results of an unsuccessful event registration.....	26
6. Sample response from a successful list tasks request.....	30
7. Example: Handlers enabled for the event type.....	31
8. Example: Returned results of a successful list handlers request.....	31
9. Example: Returned results of a successful list handlers request.....	31
10. Process for routing requests and responses between application servers.....	33
11. Sample request to retrieve wrapped data from an application server.....	38
12. Sample response for retrieving wrapped data from an application server.....	38
13. Sample request to retrieve unwrapped data from an application server.....	38
14. Sample response for retrieving unwrapped data from an application server.....	39
15. Sample request to retrieve binary data from an application server.....	39
16. Sample response for retrieving binary data from an application server.....	40
17. Sample request to update data on an application server.....	42
18. Sample response for updating data on an application server.....	42
19. Sample request to delete data from an application server.....	45
20. Sample response for deleting data from an application server.....	45
21. Cloud Provisioning Summary.....	46
22. Sample request to obtain an IP address, with the request body.....	58
23. Sample request to release an IP address, with the request body.....	60

24. Sample request to obtain a port, with request body.....	64
25. Sample request to release a port, with the request body.....	67
26. Sample request to obtain a SNA application name, with request body.....	69
27. Sample request to release a SNA application name.....	71
28. Sample request to add a classification rule.....	74
29. Sample request to remove a classification rule.....	76
30. Response from a get data set attributes request.....	78
31. Sample request to get data set attributes.....	78
32. Sample request to obtain an LPAR entry.....	100
33. Sample request to release an LPAR entry.....	102
34. Sample request to get a domain.....	109
35. Sample request to get a domain history.....	112
36. Sample request to list domains.....	116
37. Sample request to create a tenant.....	121
38. Sample request to get a tenant.....	126
39. Sample request to get a tenant history.....	128
40. Sample request to list tenants.....	134
41. Sample request to delete a tenant.....	136
42. Sample request to assign CPU properties to a tenant, with the request body.....	138
43. Sample request to assign memory capping properties to a tenant, with the request body.....	140
44. Sample request to assign a solution ID, with the request body.....	141
45. Sample request to disable CPU capping.....	143
46. Sample request to disable memory capping.....	144
47. Sample request to disable metering.....	145
48. Sample request to enable CPU capping.....	147

49. Sample request to enable memory capping.....	148
50. Sample request to enable metering.....	150
51. Sample request to add consumers to a tenant, with the request body.....	151
52. Sample request to remove tenant consumer, with the request body.....	153
53. Sample request to add tenant description, with the request body.....	155
54. Sample request to add tenant groups, with the request body.....	156
55. Sample request to remove tenant groups, with the request body.....	158
56. Sample request to get a resource pool.....	163
57. Sample request to get a domain resource pool.....	170
58. Sample request to get a resource pool history.....	173
59. Sample request to list resource pools.....	178
60. Sample request to list domain resource pools.....	185
61. Sample request to list template resource pools.....	191
62. Sample request to update security state, with the request body.....	194
63. Sample request to get security profile information.....	196
64. Sample request to create a standard software services template.....	210
65. Sample response body.....	210
66. Sample request to create a composite software services template.....	210
67. Sample response body for a composite template.....	211
68. Sample request to create a new version of a software services template.....	217
69. Sample response body.....	218
70. Sample request to create a new verison of a composite software services template.....	218
71. Sample request to create a new software services template based on an existing one, with request body.....	221
72. Sample response body.....	221
73. Sample request to modify a software services template.....	226

74. Sample request to modify a composite software services template.....	227
75. Sample request to delete a software services template.....	228
76. Sample request to retrieve a standard software services template.....	241
77. Sample request to retrieve a composite software services template.....	242
78. Sample request to retrieve a software template history.....	245
79. Sample request to retrieve software services template documentation.....	247
80. Sample request to retrieve prompt variables.....	249
81. Response body for the GET prompt variables request.....	249
82. Sample request to retrieve prompt variables, composite template.....	250
83. Response body for the GET prompt variables request, composite template.....	250
84. Sample request to retrieve source information.....	252
85. Response body for the get source request.....	252
86. Sample request to retrieve source information for a composite template.....	252
87. Response body for the get source request for a composite template.....	253
88. Sample request to list software services templates.....	257
89. Sample request to publish a software services template.....	261
90. Sample request to test a standard software services template.....	264
91. Sample response body.....	264
92. Sample request to test a composite software services template.....	264
93. Sample response body.....	265
94. Sample request to refresh a software services template.....	266
95. Sample request to archive a software services template.....	267
96. Sample request to add an approval record for a software services template.....	269
97. Sample response body.....	269
98. Sample request to get an approval record for a software services template.....	271

99. Sample response body.....	271
100. Sample request to list the approval records for a software services template.....	273
101. Sample response body.....	274
102. Sample request to approve an approval record for a software services template.....	276
103. Sample request to perform batch approval for a software services template.....	277
104. Sample request to perform batch rejection of approvals for a software services template.....	277
105. Sample request to perform batch approval and rejection of approvals for a software services template.....	278
106. Sample request to reject an approval record for a software services template.....	279
107. Sample request to delete an approval record for a software services template.....	280
108. Sample request to indicate security is complete for a software services template.....	281
109. Sample request to run a software services template.....	287
110. Sample response body.....	288
111. Sample request to test a composite software services template.....	288
112. Sample response body.....	288
113. Sample request to get a standard software services template.....	300
114. Sample request to retrieve a published software template history.....	303
115. Sample request to get consumer documentation for a software services template.....	305
116. Sample request to get the prompt variables for a published software service template.....	308
117. Sample request to retrieve prompt variables, composite template.....	309
118. Response body for the GET prompt variables request, composite template.....	310
119. Sample request to list all published software service templates.....	315
120. Response body for the GET request.....	315
121. Sample request to run a software services template.....	317
122. Sample request to create a software services instance.....	332
123. Sample request for a composite.....	333

124. Sample request to get software services instance properties.....	343
125. Sample request to list software services instances.....	350
126. Sample request to get software services instance variables.....	356
127. Sample response from a get software services instance variables request	356
128. Sample request to get software services instance variables in key-value format.....	358
129. Sample response from a get key-value variables request	358
130. Sample request to update a software services instance property.....	364
131. Sample request to update variables for a software services instance.....	366
132. Sample request to delete a software services instance.....	368
133. Sample request to resume a provisioning workflow for a software services instance.....	370
134. Sample request to perform an action against a software services instance variables.....	373
135. Sample response from a get software services instance variables request	373
136. Sample request to resume an action workflow for a software services instance.....	375
137. Sample request to retry a provisioning workflow for a software services instance.....	377
138. Sample request to retry an action workflow for a software services instance.....	379
139. Sample request to get software services instance actions.....	383
140. Sample response for performed actions	383
141. Sample request to list performed actions.....	387
142. Sample response from a list actions request	387
143. Sample request to delete a response for a performed action	389
144. Sample request to create SSINs.....	393
145. Sample request to list SSINs.....	395
146. Sample request to create a variable name.....	397
147. Sample request to create a variable name.....	399
148. Sample JSON structure for persisted data.....	402

149. Sample request to persist user-specific data.....	403
150. Sample response from a request to persist user-specific data.....	403
151. Sample JSON structure for persisted data.....	404
152. Sample request to retrieve persisted data.....	405
153. Sample response from a request to retrieve persisted data.....	405
154. Sample JSON structure for persisted data.....	406
155. Sample request to delete persisted data.....	407
156. Sample response from a request to delete persisted data.....	408
157. Example sysplex and system configuration.....	409
158. Sample request to retrieve data from one system.....	415
159. Sample response from a request to retrieve data from one system.....	415
160. Sample request to retrieve data from a list of systems.....	415
161. Sample response from a request to retrieve data from a list of systems.....	416
162. Sample request to retrieve data from all the systems in a group.....	416
163. Sample response from a request to retrieve data from all the systems in a group.....	417
164. Sample request to retrieve data from all the systems in a sysplex.....	418
165. Sample response from a request to retrieve data from all the systems in a sysplex.....	418
166. Sample request to retrieve data from all the systems in a CPC.....	419
167. Sample response from a request to retrieve data from all the systems in a CPC.....	419
168. Sample request to update data for one system.....	422
169. Sample response from a request to update data for one system.....	422
170. Sample request to update data for a list of systems.....	422
171. Sample response from a request to update data for a list of systems.....	423
172. Sample request to update data for all the systems in a group.....	423
173. Sample response from a request to update data for all the systems in a group.....	424

174. Sample request to update data for all the systems in a sysplex.....	424
175. Sample response from a request to update data for all the systems in a sysplex.....	425
176. Sample request to update data for all the systems in a CPC.....	425
177. Sample response from a request to update data for all the systems in a CPC.....	426
178. Sample request to delete data from one system.....	428
179. Sample response from a request to delete data from one system.....	428
180. Sample request to delete data from a list of systems.....	429
181. Sample response from a request to delete data from a list of systems.....	429
182. Sample request to delete data from all the systems in a group.....	429
183. Sample response from a request to delete data from all the systems in a group.....	430
184. Sample request to delete data from all the systems in a sysplex.....	430
185. Sample response from a request to delete data from all the systems in a sysplex.....	431
186. Sample request to delete data from all the systems in a CPC.....	431
187. Sample response from a request to delete data from all the systems in a CPC.....	432
188. Successful response when authenticating with a system.....	433
189. Response when the authentication request fails.....	433
190. Sample request to authenticate with a system.....	433
191. Successful response when authenticating with an HTTP proxy server.....	435
192. Sample response when the authentication request fails.....	435
193. Sample request to authenticate with an HTTP proxy server.....	435
194. List MVS subsystems.....	438
195. Sample request body to validate security resources.....	451
196. Sample response body from the validate security resources request.....	451
197. Sample request body to validate security resources.....	455
198. Sample response body from the validate security resources request.....	455

199. Sample request body to validate security resources.....	458
200. Sample response body from the provision security resources request.....	458
201. Sample request to retrieve a list of software instances.....	463
202. Sample response from a request to retrieve a list of software instances.....	463
203. Sample request to retrieve the properties of a software instance.....	469
204. Sample response from a request to retrieve the properties of a software instance.....	470
205. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server.....	471
206. Sample request to list the data sets included in a software instance.....	474
207. Sample response for a list data sets request.....	475
208. Sample request to obtain the status of a list data sets request.....	475
209. Sample get status response when the list data sets request is in progress.....	475
210. Sample get status response when the list data sets request is complete.....	475
211. Adding a software instance: request content.....	477
212. Sample request to add a software instance.....	481
213. Sample request.....	486
214. Sample response.....	487
215. Sample response.....	487
216. Sample response.....	487
217. Modifying a software instance: request content.....	490
218. Sample request to modify a software instance.....	495
219. Sample response for a modify software instance request.....	495
220. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server.....	496
221. Sample request to retrieve the product, feature, and FMID information for a software instance.....	499
222. Sample response for a retrieve product, feature, and FMID information request.....	500
223. Sample request to obtain the status of a retrieve product, feature, and FMID information request.....	500

224. Sample get status response when the retrieve product, feature, and FMID information request is in progress.....	500
225. Sample get status response when the retrieve product, feature, and FMID information request is complete.....	500
226. Sample request to delete a software instance.....	502
227. Sample response for a delete software instance request.....	502
228. Sample request to retrieve a list of portable software instances.....	509
229. Sample response from a request to retrieve a list of portable software Required authorizations....	509
230. Sample request to retrieve a list of portable software instances.....	516
231. Sample request to retrieve the properties of a portable software instance.....	524
232. Sample response from a request to retrieve the properties of a portable software instance.....	525
233. Adding a portable software instance: request content.....	526
234. Sample request to add a portable software instance.....	529
235. Response to sample request to add a portable software instance.....	529
236. Sample GET request to the status monitor URL.....	529
237. Response to sample GET request to the status monitor URL.....	529
238. Sample final request to the status monitor URL.....	529
239. Response to sample final request to the status monitor URL.....	530
240. Sample request to delete a portable software instance.....	531
241. Sample response for a delete portable software instance request.....	531
242. Sample POST request for Missing Critical Updates.....	535
243. Sample POST response for Missing Critical Updates.....	535
244. Sample GET request for Missing Critical Updates.....	535
245. Sample GET response for Missing Critical Updates.....	535
246. Sample POST request for Missing FIXCAT Updates.....	539
247. Sample POST response for Mission FIXCAT Updates.....	539
248. Sample GET request for Missing FIXCAT Updates.....	539

249. Sample GET response for Missing FIXCAT Updates.....	540
250. Sample POST request for Software Update Search.....	543
251. Sample POST response for Software Update Search.....	543
252. Sample GET request for Software Update Search.....	544
253. Sample GET response for Software Update Search.....	544
254. Sample response from a request to retrieve the z/OS system UUID.....	546
255. Sample request to get a list of storage groups.....	559
256. Sample response body.....	560
257. Sample request to get a storage-group definition.....	569
258. Sample response body.....	570
259. Sample request to get a list of volumes.....	575
260. Sample response body.....	576
261. Sample request to get a volume definition.....	579
262. Sample response body.....	580
263. Sample request to get a list of data classes with summary information about each data class.....	596
264. Sample response body.....	597
265. Sample request to get a list of data classes with detailed information about each data class.....	612
266. Sample response body.....	613
267. Sample request to get a list of storage classes with summary information about each storage class.....	620
268. Sample response body.....	621
269. Sample request to get a list of storage classes with detailed information about each storage class.....	627
270. Sample response body.....	628
271. Sample request to define a volume list to a storage group without invoking activation or validation.....	632
272. Sample response body when adding volumes is successful.....	632

273. Sample response body when adding volumes fails.....	632
274. Sample request to define a volume list to a storage group and validate the SCDS.....	633
275. Sample response body when adding volumes is successful and validation is successful.....	633
276. Sample response body when adding volumes is successful but validation fails.....	633
277. Sample request to define a volume list to a storage group.....	633
278. Sample response body when adding volumes and activation is successful.....	634
279. Sample response body when the activation result is not available.....	634
280. Sample response body when validation or activation fails.....	634
281. Sample success response body.....	637
282. Sample failed response body.....	637
283. Sample inconclusive response body.....	637
284. Sample success response body.....	639
285. Sample response body.....	639
286. Sample response from a request to retrieve a list of the currently defined administrative CFRM policies.....	661
287. Sample response from a request to retrieve a list of the CFRM policy properties.....	667
288. Sample request to activate a specified CFRM policy.....	668
289. Sample request to rename an existing CFRM policy to a new name.....	669
290. Sample request to modify the properties of a CFRM policy.....	676
291. Sample request to add a CFRM policy to an administrative policy.....	683
292. Sample request to copy an existing policy with a new name.....	685
293. Sample request to delete a CFRM policy from an administrative policy.....	686
294. Sample request to delete multiple CFRM policies from an administrative policy.....	686
295. Sample request to retrieve a list of systems.....	691
296. Sample response from a request to retrieve a list of systems.....	692
297. Sample request to retrieve a list of groups.....	693

298. Sample response from a request to retrieve a list of groups.....	694
299. Sample request to retrieve a list of systems included in a group.....	696
300. Sample response from a request to retrieve a list of systems included in a group.....	696
301. Sample request to retrieve a list of sysplexes.....	698
302. Sample response from a request to retrieve a list of sysplexes.....	698
303. Sample request to retrieve a list of systems included in a sysplex.....	700
304. Sample response from a request to retrieve a list of systems included in a sysplex.....	700
305. Sample request to retrieve a list of systems included in a CPC.....	702
306. Sample response from a request to retrieve a list of systems included in a CPC.....	703
307. Sample request to check systems selected.....	706
308. Sample response to check systems selected.....	707
309. Sample request to Check all the MVS Systems in the local plex.....	711
310. Sample response to Check all the MVS Systems in the local plex.....	712
311. Sample request to create a new TSO/E address space on the local system.....	720
312. Sample response from create TSO/E address space request on the local system.....	720
313. Sample request to create a new TSO/E address space on a remote system.....	721
314. Sample response from create TSO/E address space request on a remote system.....	721
315. Sample request to reconnect to an existing TSO/E address space on the local system	721
316. Sample response from a reconnect to TSO/E address space request on the local system.....	721
317. Sample request to reconnect to an existing TSO/E address space on a remote system.....	722
318. Sample response from a request to reconnect to a TSO/E address space on a remote system.....	722
319. Starting an application: example of the request content.....	723
320. Sample request to start an application in a TSO/E address space.....	724
321. Sample response from a start an application in a TSO/E address space request.....	725
322. Sample request to send a message to a TSO/E address space.....	738

323. Sample response from send message to TSO/E address space request.....	739
324. Sample request to send a message to an application.....	740
325. Sample response from send message to an application request.....	740
326. Sample request to ping a TSO/E address space.....	741
327. Sample response from ping TSO/E address space request.....	741
328. Sample request to receive a message from a TSO/E address space.....	742
329. Sample response from receive message from a TSO/E address space request.....	743
330. Sample request to receive messages from an application.....	744
331. Sample response for request to receive messages from an application.....	744
332. Sample request to logoff a TSO/E address space.....	745
333. Sample response for logoff a TSO/E address space request.....	745
334. Sample request to cancel a TSO/E address space.....	746
335. Sample response for a cancel TSO/E address space request.....	746
336. Sample request to issue a prime WRP request.....	749
337. Sample response body.....	749
338. Sample request to issue a delete WLM resource pool request.....	750
339. Sample response body.....	750
340. Sample request to construct a service definition based on the current installed definition.....	752
341. Sample response body.....	752
342. Sample request to construct a service definition by removing the classification rule, then installing the new service definition.....	754
343. Sample response body.....	754
344. Sample request to get metered data.....	757
345. Sample response.....	758
346. Sample request to get DDS server time data.....	759
347. Sample response.....	760

348. Sample request to post the request for compliance.....	763
349. Sample response from post the request for compliance.....	763
350. Sample request to issue a command synchronously.....	780
351. Sample response body.....	780
352. Sample request to issue a command and detect a keyword.....	781
353. Sample response body.....	781
354. Sample request to issue a system command asynchronously.....	781
355. Sample response body.....	781
356. Sample request to issue an s PEGASUS command synchronously and detect the keyword PEGASUS.....	781
357. Sample response body.....	781
358. Sample request to issue an s PEGASUS command asynchronously and detect the keyword XIAOX.....	782
359. Sample response body.....	782
360. Sample request to get the response for a system command that was issued asynchronously.....	788
361. Sample response body for a get command response request.....	788
362. Sample request to get the detect result.....	794
363. Sample response body for a get detect result request.....	794
364. Sample request to get the detect result.....	794
365. Sample response body for a successful get detect result request.....	795
366. Sample request to get messages from a 1-hour duration.....	800
367. Sample response body for a Get Messages request.....	801
368. Sample request for a Get Messages request that starts at a specific timestamp.....	801
369. Sample response body for a Get Messages request.....	801
370. Sample request to create system variables.....	805
371. Sample request body for a create system variables request.....	805
372. Sample request to get system variables.....	808

373. Sample response from a get system variables request.....	808
374. Sample request to get system symbols.....	808
375. Sample response from a get system symbols request.....	808
376. Sample request to get 2 system variables.....	808
377. Sample response from a get system symbols request.....	808
378. Sample request to get 2 system symbols.....	808
379. Sample response from a get system symbols request.....	809
380. Sample request to import system variables.....	810
381. Sample request body for an import system variables request.....	810
382. Sample request to export system variables.....	811
383. Sample request body for an export system variables request.....	812
384. Sample request to delete system variables.....	813
385. Sample request body for a delete system variables request.....	813
386. Example: list all of the data sets.....	822
387. Example: List all of the cataloged data sets with specified base attributes.....	823
388. Example: List all of the members of a data set.....	825
389. Example: List all of the members of a data set with specified base attributes.....	826
390. Example: Returned contents of the SMFPRM00 member of sys1.parmlib.....	830
391. Example: Returned contents of the SMFPRM00 member of sys1.parmlib.....	830
392. Example: Retrieve the contents of a sequential data set.....	831
393. Example: Request header for a write request to the SMFPRM00 member of sys1.parmlib.....	835
394. Example: Request body for a write request to the SMFPRM00 member of sys1.parmlib.....	835
395. Example: Contents of a sequential data set.....	836
396. Example: Contents of a member of data set with regular expression.....	836
397. Example: Create a data set	839

398. Example: Create data set.....	839
399. Example: Create PDSE data set.....	840
400. Example: Delete a data set	841
401. Example: Delete uncatalogued data set.....	842
402. Example: Delete a member of a cataloged partitioned data set.....	843
403. Example: Delete a member of an uncataloged partitioned data set.....	844
404. 'rename' request.....	844
405. 'copy' request.....	844
406. 'hsmigrate', 'hrecall', or 'hdelete' request.....	844
407. Example: Rename MY.OLD.DSN to MY.NEW.DSN.....	848
408. Example: copy member MYMEM1 from MY.OLD.DSN to MY.NEW.DSN(MYMEM2).....	848
409. IDCAMS Access Methods Services.....	850
410. Example: Returned list of UNIX files and directories in path /usr.....	854
411. Example: Returned list of UNIX files	854
412. Example: List UNIX files with more query parameters.	855
413. Example: Response body for a GET request to the UNIX file /etc/inetd.conf.....	859
414. Example: Request body for a PUT request to the UNIX file /etc/inetd.conf.....	862
415. Example: Create a UNIX file.....	864
416. Example: Create a UNIX directory.....	865
417. Example: Delete a UNIX file.....	866
418. Example: Delete a UNIX directory.....	867
419. Example: Rename a UNIX file.....	873
420. List UNIX Filesystems.....	875
421. Create UNIX Filesystems.....	877
422. Delete UNIX Filesystems.....	879

423. Example: Mount a UNIX file system.....	881
424. Example: Unmount a UNIX file system.....	883
425. Format of resource URLs for z/OS jobs REST interface.....	902
426. Example: Returned job status.....	908
427. Example: Returned status for an active step.....	908
428. Example: Returned list of the jobs for a specific owner and job name prefix.....	912
429. Example: Returned list of spool files.....	914
430. Example: Returned spool file content.....	917
431. Example: Returned spool file content (a range of records).....	918
432. Example: Returned job content (the job JCL).....	918
433. Example: Returned results of a job submission.....	923
434. Example: Returned results of a job hold request.....	926
435. Example: Returned results of a job release request.....	928
436. Example: Returned results of a job class change.....	931
437. Example: Returned results of a job cancellation.....	933
438. Example: Results of a job delete request.....	936
439. Sample request to get the list of services in z/OS Management Services Catalog.....	958
440. Sample request to get the details of a specific service in the z/OS Management Services catalog..	973
441. Sample request to get the list of categories defined to z/OS Management Services.....	977
442. Sample request to get the details of a specific category defined to z/OS Management Services.....	978
443. Sample request to get the list of service submissions in z/OS Management Services Catalog.....	981
444. Sample request to create a new service submission in the z/OS Management Services Catalog.....	987
445. Sample request to get the details of a specific service submission in the z/OS Management Services.....	987
446. Sample request to delete a service submission from the z/OS Management Services Catalog.....	990
447. Sample request to get the details of a specific service submission in the z/OS Management Services.	994

448. Sample request to update a service submission from the z/OS Management Services Catalog.....	997
449. Sample request to run a service submission.....	999
450. Sample request to get the list of job statements that are defined in the z/OS Management Services Catalog global settings.....	1000
451. Sample request to get the list of target systems.....	1002
452. Sample request to retrieve z/OSMF information.....	1005
453. Sample request to persist user-specific z/OSMF settings data.....	1009
454. Sample request to retrieve z/OSMF settings data.....	1010
455. Sample response from a request to retrieve z/OSMF settings data.....	1011
456. Sample request to delete persisted z/OSMF settings data.....	1012
457. Sample response from a request to retrieve z/OSMF settings data.....	1012
458. Sample request to retrieve z/OSMF settings data.....	1012
459. Sample response from a request to retrieve z/OSMF settings data.....	1012
460. Log in and obtain a JSON Web Token for user authentication.....	1016
461. Change a user password.....	1018
462. Delete the JSON Web Tokens for this user.....	1020
463. Sample request to create a workflow.....	1031
464. Sample response from a create workflow request.....	1031
465. Sample request to get workflow properties.....	1050
466. Sample response from a get workflow properties request (Part 1 of 3).....	1051
467. Sample response from a get workflow properties request (Part 2 of 3).....	1052
468. Sample response from a get workflow properties request (Part 3 of 3).....	1053
469. Sample request to list workflows.....	1056
470. Sample response from a list workflows request.....	1056
471. Sample request to start a workflow.....	1061
472. Sample response from a start workflow request.....	1061

473. Sample request to cancel a workflow.....	1063
474. Sample response from a cancel workflow request.....	1063
475. Sample request to delete a workflow.....	1064
476. Sample response from a delete workflow request.....	1065
477. Sample request to get a workflow definition.....	1075
478. Sample response from a get workflow definition request (Part 1 of 3).....	1075
479. Sample response from a get workflow definition request (Part 2 of 3).....	1076
480. Sample response from a get workflow definition request (Part 3 of 3).....	1077
481. Sample request to archive a workflow.....	1079
482. Sample response from an archive workflow request.....	1079
483. Sample request to list archived workflows.....	1081
484. Sample response from a list archived workflows request.....	1082
485. Sample request to get archived workflow properties.....	1095
486. Sample response from a get archived workflow properties request.....	1096
487. Sample request to delete an archived workflow.....	1097
488. Sample response from a delete archived workflow request.....	1097
489. General metadata for a workflow.....	1111
490. You can define various questions for step owners to answer.....	1116
491. How feedback questions can be included in the steps in your workflow.....	1116
492. This sample step submits a REXX exec for immediate processing.....	1128
493. Sample REST step definition with substitution variables and property mapping variables.....	1136
494. Example: Defining a called workflow on the step element tag.....	1143
495. You can designate a step as automated by adding the autoEnable element to the <step> element tag.....	1145
496. Example of a suspend step.....	1146
497. You can designate a step as conditional by adding the condition element to the <step> element tag.....	1149

498. You can use variable values in the condition to be satisfied.....	1149
499. Specifying attributes on the variable element.....	1158
500. Variable definition in this example.....	1161
501. How the atCreate element is used to specify variable attributes for required and prompt.....	1161
502. Specifying the use of the simplified instance variable format in a workflow definition	1162
503. You can use variable values in the condition to be satisfied.....	1163
504. Example of how a system variable can be used in a workflow step.....	1166
505. Example of the returned object from the Get system variables service.....	1167
506. Variable substitution for a system variable in the Workflows task customize job statements window.....	1167
507. Results of variable substitution for a system variable in the Workflows task.....	1167
508. Example of how an array variable might be used in a workflow step.....	1168
509. Example of how an array variable might be used in a workflow step.....	1169
510. Format of a workflow variable input file.....	1170
511. Example of a workflow variable input file.....	1170
512. Using variable substitution in a workflow variable input file.....	1171
513. How variable substitution in the workflow variable input file works with the Workflows task Step Perform wizard.....	1172
514. Syntax to use to call the programmaticallyCloseTab function.....	1235
515. Sample code for the programmaticallyCloseTab function.....	1236
516. Syntax to use to call the getUserSessionId function.....	1236
517. Sample code for the getUserSessionId function.....	1236
518. Syntax to use to call the cleanupBeforeDestroyComplete function.....	1236
519. Sample code for the cleanupBeforeDestroyComplete function.....	1237
520. Sample JavaScript code for instantiating the zosmfExternalTools object	1237
521. Syntax to use for the isContentChanged function.....	1237
522. Sample confirmation window for a close task request.....	1238

523. Sample code for the isContentChanged function.....	1238
524. Syntax to use for the shouldClose function.....	1239
525. Syntax to use for the cleanupBeforeDestroy function.....	1240
526. Sample code for the cleanupBeforeDestroy function.....	1240
527. Sample code for importing and instantiating the Applinker API.....	1241
528. Syntax to use to call the sendEvent function.....	1243
529. Sample code for the sendEvent function.....	1244
530. Syntax to use to call the getHandlers function.....	1244
531. Syntax to use to call the hasLaunchContext function.....	1245
532. Sample code for the hasLaunchContext function.....	1246
533. Syntax to use to call the getEventFromUrl function.....	1246
534. Sample code for the getEventFromUrl function.....	1248
535. Syntax to use to call the subscribe function.....	1249
536. Sample code for the subscribe function.....	1249
537. Syntax to use to call the onLoadingComplete function.....	1250
538. Sample code for creating a log prefix.....	1251
539. Sample code for setting up client side logging.....	1252
540. Sample z/OSMF client side log data.....	1253
541. Sample request to retrieve a file.....	1256
542. Sample response for a request to retrieve a file.....	1256
543. Screen capture of the z/OSMF help system.....	1257
544. Table of contents template for panel help plug-ins.....	1263
545. Sample table of contents for the System Status task.....	1264
546. Table of contents template for message help plug-ins.....	1264
547. Sample table of contents for the messages issued by z/OSMF core.....	1265

548. Sample code for linking to a help topic.....	1268
549. Sample plug-in property file.....	1273
550. Bootstrap properties for z/OSMF.....	1284
551. Sample product information file for the Software Management task.....	1287
552. Using the Add software instance REST API to specify a data set property in the request content.	1302
553. Sample entry in the list for the subject data set.....	1303
554. Sample template step to determine the specified property.....	1303

Tables

1. SAF identifiers for the z/OSMF REST interfaces.....	7
2. Operations provided through the Application Linking Manager interface services.....	9
3. Event types shipped with z/OSMF.....	12
4. Event requestors shipped with z/OSMF.....	16
5. Event handlers shipped with z/OSMF.....	21
6. Valid applId values for the z/OSMF plug-ins.....	27
7. Valid taskId values for the z/OSMF tasks.....	27
8. Operations provided through the application server routing services.....	34
9. Supported input parameters for the application server routing services.....	36
10. Supported input parameters for the application server routing services.....	41
11. Supported input parameters for the application server routing services.....	43
12. Resources for Cloud Provisioning.....	48
13. User roles for Cloud Provisioning.....	48
14. Objects for Cloud Provisioning.....	49
15. SAF resources for Cloud Provisioning Roles.....	49
16. z/OSMF resource pool services: operations summary.....	51
17. Request content for the obtain IP address request.....	53
18. Network parameters fields.....	53
19. Default values for recovery-method.....	55
20. Systems field.....	56
21. HTTP error response codes for an obtain IP address request.....	57
22. Response from an obtain IP address request.....	57
23. Request content for the release IP address request.....	59

24. Network parameters fields.....	59
25. Systems field.....	59
26. HTTP error response codes for a release IP address request.....	60
27. Request content for the obtain port request.....	61
28. Network parameters fields.....	62
29. Systems field.....	63
30. HTTP error response codes for an obtain port request.....	63
31. Response from an obtain port request.....	64
32. Request content for the release port request.....	65
33. Network parameters fields.....	66
34. Systems field.....	66
35. HTTP error response codes for a release port request.....	66
36. Request content for the obtain SNA application name request.....	68
37. Network parameters fields.....	68
38. HTTP error response codes for an obtain SNA application name request.....	69
39. Response from an obtain SNA application name request.....	69
40. Request content for the release SNA application name request.....	70
41. Network parameters fields.....	71
42. HTTP error response codes for a release SNA application name request.....	71
43. Request content for the add classification rule request.....	72
44. WLM parameters fields.....	73
45. HTTP error response codes for a release SNA application name request.....	73
46. Response from an add classification rule request.....	73
47. Request content for the remove classification rule request.....	75
48. WLM parameters fields.....	75

49. HTTP error response codes for a release SNA application name request.....	75
50. HTTP error response codes for a get data set attributes request.....	77
51. Request content for the modify LPAR entry request.....	79
52. z/OS volumes object.....	81
53. NJE node.....	81
54. HTTP error response codes for a create an LPAR resource pool entry request.....	82
55. Response from a create an LPAR resource pool entry request.....	82
56. Request content for the modify LPAR entry request.....	83
57. z/OS volumes object.....	85
58. NJE node.....	86
59. HTTP error response codes for a modify LPAR resource pool entry request.....	86
60. HTTP error response codes for a delete an LPAR resource pool entry request.....	87
61. HTTP error response codes for a list LPAR entries request.....	88
62. Response from a list LPAR entries request.....	88
63. z/OS volumes object.....	91
64. NJE node.....	91
65. HTTP error response codes for a retrieve LPAR entry properties request.....	92
66. Response from a retrieve LPAR entry properties request.....	92
67. z/OS volumes object.....	95
68. NJE node.....	95
69. Request content for the obtain LPAR entry request.....	95
70. LPAR parameter fields.....	96
71. HTTP error response codes for an obtain LPAR resource pool entry request.....	96
72. Response from an obtain LPAR entry properties request.....	97
73. z/OS volumes object.....	99

74. NJE node.....	99
75. Request content for the release LPAR entry request.....	100
76. LPAR parameter fields.....	101
77. HTTP error response codes for a release LPAR resource pool entry request.....	101
78. z/OSMF resource management services: operations summary.....	102
79. HTTP error response codes for a get domain request.....	105
80. Response from a get domain request.....	105
81. Response from a get request: Systems.....	108
82. Response from a create request: SAF-resource object.....	109
83. HTTP error response codes for a get domain history request.....	111
84. Response from a get request: History object.....	111
85. HTTP error response codes for a list domains request.....	113
86. Response from a list domains request.....	113
87. Properties of domains.....	113
88. Request content for the create tenant request.....	118
89. Tenant metering and capping properties object.....	118
90. Tenant managed by request object.....	119
91. Systems request object.....	120
92. HTTP error response codes for a get domain request.....	120
93. Response from a create tenant request.....	120
94. HTTP error response codes for a get tenant request.....	122
95. Response from a get tenant request.....	122
96. Response from a get request: tenant-templates.....	125
97. Response from a create request: SAF-resource object.....	125
98. HTTP error response codes for a get tenant history request.....	127

99. Response from a get request: History object.....	128
100. HTTP error response codes for a list tenants request.....	129
101. Response from a list tenants request.....	130
102. Tenant list.....	130
103. HTTP error response codes for a delete tenant request.....	136
104. Request content for the assign CPU properties to a tenant request.....	137
105. HTTP error response codes for an assign CPU properties to a tenant request.....	138
106. Request content for the assign memory capping properties to a tenant request.....	139
107. HTTP error response codes for an assign CPU properties to a tenant request.....	139
108. Request content for the assign solution ID request.....	140
109. HTTP error response codes for an assign solution ID request.....	141
110. HTTP error response codes for a disable CPU capping request.....	142
111. HTTP error response codes for a disable memory capping request.....	144
112. HTTP error response codes for a disable metering request.....	145
113. HTTP error response codes for an enable CPU capping request.....	146
114. HTTP error response codes for an enable memory capping request.....	148
115. HTTP error response codes for an enable metering request.....	149
116. Request content for the add tenant consumer request.....	151
117. HTTP error response codes for an add tenant consumer request.....	151
118. Request content for the remove tenant consumer request.....	152
119. HTTP error response codes for a remove tenant consumer request.....	152
120. Request content for the add tenant description request.....	154
121. HTTP error response codes for an add tenant description request.....	154
122. Request content for the add tenant groups request.....	155
123. HTTP error response codes for an add tenant groups request.....	156

124. Request content for the remove tenant groups request.....	157
125. HTTP error response codes for a remove tenant groups request.....	157
126. HTTP error response codes for a get resource pool request.....	159
127. Response from a get resource pool request.....	159
128. Response from a get request: Systems.....	163
129. HTTP error response codes for a get domain resource pool request.....	165
130. Response from a get domain resource pool request.....	165
131. Response from a get request: Systems.....	169
132. HTTP error response codes for a get resource pool request.....	172
133. Response from a get request: History object.....	172
134. HTTP error response codes for a get resource pool request.....	174
135. Response from a list resource pool request.....	174
136. Resource pool list.....	174
137. HTTP error response codes for a get domain resource pool request.....	180
138. Response from a list domain resource pool request.....	180
139. Domain resource pool list.....	180
140. HTTP error response codes for a get template resource pool request.....	186
141. Response from a list template resource pool request.....	187
142. Template resource pool list.....	187
143. Request content for the update security state request.....	193
144. HTTP error response codes for an update security state request.....	193
145. HTTP error response codes for a get security resources request.....	194
146. Response from a get security resources request.....	195
147. SecurityValidationItem structure.....	195
148. z/OSMF software services template services: operations summary.....	197

149. Response from a software services template request failure.....	202
150. Request content for a request to create a software services template.....	204
151. Composite-definition structure.....	207
152. Connector object.....	208
153. Response from a create software services template request.....	208
154. Response from a create request: SAF-resource object.....	209
155. Response from a software services template request failure.....	209
156. Request content for the software services template request.....	212
157. Composite-definition structure.....	215
158. Connector object.....	216
159. Response from a create new version of a software services template request.....	216
160. Response from a create request: SAF-resource object.....	217
161. Request content for the software services template request.....	219
162. Response from a successful request.....	220
163. Response from a create request: SAF-resource object.....	220
164. Request content for a request to modify a software services template.....	222
165. Composite-definition structure.....	225
166. Connector object.....	226
167. Response from the software services template request.....	228
168. Composite-affected object.....	228
169. Response from a request to get a software services template.....	229
170. Response from a get request: Approval-Object.....	235
171. Response from a get request: Action-Object.....	236
172. Response from a get request: Variable-Object.....	237
173. Response from a get request: Prompt-Variable-Object.....	237

174. Response from a create request: SAF-resource object.....	238
175. Response from a get request: RunAsUser object.....	238
176. Response from a get request: Composite-definition object.....	239
177. Connector object.....	240
178. Response from a get request: Security Workflow Information.....	241
179. Response from a get request: History object.....	245
180. Response from a get prompt variables request.....	248
181. Response from a get request: Prompt-Variable-Object	248
182. Response from a get request: Composite-Prompt-Variable-Object.....	249
183. Response from a get source request.....	251
184. Response from a get request: Source-Info-Object.....	252
185. Array of objects.....	254
186. Fields for each software services template.....	254
187. Contents of composite-definition object.....	257
188. Request content for the software services template request.....	260
189. Response from the software services template request.....	261
190. Composite-affected object.....	261
191. Request content for the test software services template request.....	262
192. Runtime properties	263
193. Response from a test software services template request.....	263
194. Composite child registry information objects.....	264
195. Response from the software services template request.....	267
196. Composite-affected object.....	267
197. Request content for the add approval request.....	268
198. Response from an add approval request.....	269

199. Response from a get approval request.....	270
200. Response from a list approvals request.....	272
201. Response from a get approval request.....	273
202. Request content for the software services template request.....	275
203. Request content for the software services template request.....	276
204. Approval objects.....	277
205. Request content for the software services template request.....	278
206. z/OSMF published software service template services: operations summary.....	283
207. Response from a software services template request failure.....	284
208. Request content for the run software services template request.....	285
209. Runtime properties	286
210. Response from a run software services template request.....	287
211. Composite child registry information objects.....	287
212. Response from a get software services template request.....	291
213. Response from a get request: Prompt-Variable Object.....	295
214. Response from a get request: Approval-Object.....	296
215. Response from a get request: Action-Object.....	297
216. Response from a get request: Variable-Object.....	297
217. Response from a create request: SAF-resource object.....	298
218. Response from a get request: composite-definition object.....	298
219. Connector object.....	299
220. Response from a get request: RunAsUser object.....	300
221. Response from a get request: History object.....	303
222. Response from a get prompt variables request.....	307
223. Response from a get request: Prompt-Variable Object.....	307

224. Response from a get request: Composite-Prompt-Variable Object.....	308
225. Array of objects.....	312
226. Fields for each software services template.....	312
227. Contents of composite-definition object.....	314
228. Response from a software services template request failure.....	315
229. Request content for the modify software services template request.....	316
230. z/OSMF software services instance services: operations summary.....	319
231. Response from a request failure.....	320
232. Request content for a create software services instance request.....	322
233. Action structure for a "create a software services instance" request.....	326
234. Variable structure.....	327
235. Response from a get request: Prompt-Variable-Object.....	327
236. Composite-data structure.....	328
237. Connector object.....	329
238. HTTP error response codes for a create software services instance request.....	330
239. Response from a create software services instance request.....	330
240. Response from a request failure.....	330
241. HTTP error response codes for a get software services instance contents request.....	334
242. JSON object that is returned for a get software services instance property request.....	335
243. Action structure for a "get the contents of a software services instance" request	339
244. Variable structure.....	341
245. Response from a get request: Prompt-Variable-Object.....	341
246. Composite-instance data.....	342
247. Composite child object.....	342
248. Response from a request failure.....	343

249. HTTP error response codes for a list software services instances request.....	346
250. JSON object that is returned for a list software services instances request.....	347
251. Composite child object.....	349
252. Response from a request failure.....	349
253. HTTP error response codes for a get software services instance variables request.....	354
254. Get variables request: Format of the variables object.....	355
255. Variable structure.....	355
256. Response from a request failure.....	355
257. HTTP error response codes for a get software services instance key-value variables request.....	357
258. Get key-value variables request: Format of the variables object.....	358
259. Response from a request failure.....	358
260. Request content for an update software services instance request.....	359
261. Action structure for an "update software services instance" request.....	361
262. Variable structure.....	362
263. Response from a get request: Prompt-Variable-Object.....	362
264. HTTP error response codes for a update software services instance request.....	364
265. Response from a request failure.....	364
266. Request content for the update software services instance variables request.....	365
267. Variable structure.....	365
268. HTTP error response codes for a update software services instance request.....	366
269. Response from a request failure.....	366
270. HTTP error response codes for a delete software services instance request.....	367
271. Response from a request failure.....	368
272. HTTP error response codes for a resume provisioning workflow request.....	369
273. Response from a request failure.....	370

274. Request content for the perform action software services instance request.....	371
275. Input variable structure.....	371
276. HTTP error response codes for a do action request.....	372
277. Response body for the do action request.....	372
278. Response from a request failure.....	372
279. HTTP error response codes for a resume action workflow request.....	374
280. Response from a request failure.....	375
281. HTTP error response codes for a retry provisioning workflow request.....	376
282. Response from a request failure.....	377
283. HTTP error response codes for a retry action workflow request.....	378
284. Response from a request failure.....	379
285. HTTP error response codes for a get software services instance contents request.....	380
286. JSON object that is returned for a get actions request.....	381
287. Response from a request failure.....	382
288. HTTP error response codes for a "get software services instance contents" request.....	386
289. JSON object that is returned for a list actions request.....	386
290. Action object for a list actions request.....	386
291. Response from a request failure.....	387
292. HTTP error response codes for a delete action response request.....	388
293. Response from a request failure.....	389
294. SSIN services: operations summary.....	390
295. Response from a request failure.....	390
296. Request content for the create SSIN request.....	392
297. HTTP error response codes for a create SSIN request.....	392
298. Response from a create SSIN request.....	392

299. Fields in the ssin-list array.....	392
300. Response from a request failure.....	392
301. HTTP error response codes for a list SSIN request.....	394
302. Response from a list SSINs request.....	395
303. Fields in the ssin-list array.....	395
304. Response from a request failure.....	395
305. Request content for the create variable name request.....	396
306. HTTP error response codes for a create variable request.....	397
307. Response from a create variable name request.....	397
308. Response from a request failure.....	397
309. Request content for the create unique variable names request.....	398
310. HTTP error response codes for a create variable request.....	398
311. Response from a create unique variable names request.....	399
312. Response from a request failure.....	399
313. Operations provided through the data persistence services.....	400
314. Operations provided through the multisystem routing services.....	410
315. Supported input parameters for the multisystem routing services.....	413
316. Supported input parameters for the multisystem routing services.....	420
317. Supported input parameters for the multisystem routing services.....	427
318. Subsystem services method.....	435
319. Response.....	437
320. Notification methods.....	438
321. Response content from the notifications received by the current user.....	439
322. Request content for the send notification request.....	441
323. Response content for the send notification request.....	442

324. Request content from a notification that requires user input.....	444
325. Response content from a notification that requires user input.....	444
326. Operations provided through the SCA services.....	446
327. Request content to validate security resources in a post body.....	448
328. Response content to validate security resources in a post body.....	449
329. Request content used to validate security resources in a descriptor file.....	452
330. JSON security descriptor fields as indicated by the path field.....	452
331. Response content to validate security resources in a security descriptor file.....	453
332. Request content used to provision security resources defined in an HTTP request body.....	456
333. Response content to provision security resources defined in an HTTP request body.....	456
334. Operations provided through the software management services.....	458
335. Operations provided through the storage management services.....	547
336. Query parameters for retrieving a list of storage groups.....	549
337. Response content for a successful retrieval of storage-group summary information.....	552
338. Response content for a successful retrieval of storage-group detailed information.....	553
339. Query parameters for retrieving a storage group definition.....	560
340. Response content for a successful retrieval of storage-group summary information.....	561
341. Response content for a successful retrieval of storage-group detailed information.....	563
342. Query parameters for retrieving a list of volumes.....	571
343. Response content for a successful retrieval of volume information.....	573
344. Query parameter for retrieving a volume definition.....	576
345. Response content for a successful retrieval of volume information.....	577
346. Query parameters for retrieving a list of data classes.....	581
347. Response content for a successful retrieval of data classes summary information.....	583
348. Response content for a successful retrieval of data classes detailed information.....	595

349. Query parameters for retrieving a data class definition.....	598
350. Response content for a successful retrieval of data classes summary information.....	598
351. Response content for a successful retrieval of data classes detailed information.....	611
352. Query parameters for retrieving a list of storage classes.....	614
353. Response content for a successful retrieval of storage classes summary information.....	616
354. Response content for a successful retrieval of storage classes detailed information.....	620
355. Query parameters for retrieving a storage class definition.....	622
356. Response content for a successful retrieval of storage classes summary information.....	622
357. Response content for a successful retrieval of storage classes detailed information.....	626
358. Query parameters for retrieving a list of volumes.....	629
359. Request content for the define volume list for storage group request.....	630
360. Query parameters for retrieving a list of volumes.....	635
361. Request content for the validate SCDS request.....	635
362. Query parameters to activate an SCDS request.....	636
363. Query parameters to activate an SCDS request.....	638
364. Error categories for storage management API services operations.....	639
365. Category 1 errors for storage management API services operations.....	640
366. Category 2 errors for storage management API services operations.....	643
367. Category 3 errors for storage management API services operations.....	644
368. Category 4 errors for SCDS modification and validation error.....	645
369. Category 5 errors for activating SCDS action.....	657
370. Category 6 errors for security check error.....	657
371. Operations provided through the sysplex management services.....	658
372. Errors for sysplex management API services operations.....	686
373. Operations provided through the topology services.....	687

374. Validate Status Query Parameters.....	704
375. Validate System Json Responses.....	705
376. SYS1 zOSMF Local System	707
377. SYS2 zOSMF2.....	708
378. SYS3.....	708
379. SYS4 zOSMF3.....	708
380. SYS5.....	708
381. Query parameters to check all systems in the local plex.....	709
382. Validate System Json Responses.....	711
383. SYS1 zOSMF Local System	712
384. SYS2 zOSMF2.....	713
385. SYS3.....	713
386. SYS4 zOSMF3.....	713
387. SYS5.....	713
388. Operations provided through the TSO/E address space services.....	714
389. Supported parameters for the start and reconnect TSO/E session requests.....	718
390. Supported parameters for the TSO/E command with z/OSMF REST API.....	725
391. Response content for a successful issue command request.....	728
392. Response JSON object for an unsuccessful issue command request.....	729
393. Response codes for unsuccessful issue command requests.....	729
394.	734
395. Response content for a successful get response request.....	735
396. Response JSON object for an unsuccessful issue command request.....	736
397. Operations provided through the WLM resource pooling services.....	746
398. Request content for the prime WLM resource pool request.....	748

399. Response content for a successful prime WLM resource pool request.....	748
400. Response content for a delete WLM resource pool request.....	750
401. Request content for the construct a WLM service definition request.....	751
402. Response content for a successful construct a WLM service definition request.....	752
403. Request content for the construct a WLM service definition request.....	753
404. Response content for a successful construct a WLM service definition request.....	754
405. Operations provided through the RMF services.....	754
406. Query parameters.....	755
407. HTTP error response codes for a get metered data request.....	757
408. HTTP error response codes for a get DDS server time data request.....	759
409. Operations that are provided through the z/OS compliance REST interface services.....	760
410. Supported input parameters for the z/OS compliance REST service.....	762
411. Operations provided through the z/OS console services.....	763
412. Request content for the issue command request.....	766
413. HTTP error response codes for an issue command request.....	770
414. Response content for a successful synchronous issue command request.....	777
415. Response content for a successful asynchronous issue command request.....	778
416. Response content for an unsuccessful issue command request.....	780
417. HTTP error response codes for a get command response request.....	783
418. Response content for a successful get command response request.....	787
419. Response content for an unsuccessful get command response request.....	788
420. HTTP error response codes for a detect result for unsolicited messages request.....	789
421. Response content for a successful get detect result request.....	794
422. Response content for an unsuccessful get detect result request.....	794
423. Query parameters for a Get Messages request.....	795

424. HTTP Error response codes for a Get Messages request.....	798
425. Response content for a successful Get Messages request.....	799
426. Messages JSON object.....	799
427. Response content for an unsuccessful Get Messages request.....	800
428. z/OS system variable services: operations summary.....	802
429. z/OSMF system variable services: URI path variables.....	802
430. Request content for the create or update system variables request.....	804
431. Fields in a JSON object for the create or update system variables request.....	804
432. HTTP error response codes for a create or update system variables request.....	804
433. Request Parameters.....	806
434. HTTP error response codes for a get system variables request.....	807
435. Get system variables request response body.....	807
436. Get system variables request: objects.....	807
437. Request content for the import system variables request.....	809
438. HTTP error response codes for a create system variables request.....	810
439. Request content for the export system variables request.....	811
440. HTTP error response codes for a create system variables request.....	811
441. HTTP error response codes for a create system variables request.....	812
442. Operations provided through the z/OS data set and file REST interface services.....	813
443. Request body to create a sequential or partitioned data set	837
444. Request.....	845
445. Input Content.....	849
446. Response.....	849
447. Filter parameters for a "List UNIX Files and Directories" request.....	851
448. Tree traversal parameters for a "List UNIX Files and Directories" request.....	853

449. Request body to create a UNIX file or directory	863
450. JSON request document.....	868
451. Input Content.....	876
452. Request body to mount a UNIX file system.....	880
453. Request body to unmount a UNIX file system	882
454. Contents of the JSON data set list document.....	884
455. X-IBM-Attributes=vol.....	885
456. X-IBM-Attributes=dsname.....	885
457. X-IBM-Attributes=base.....	885
458. Items key:value pairs.....	885
459. Contents of the JSON PDS/PDSE member list document.....	887
460. X-IBM-Attributes=base and data set RECFM=F or V.....	887
461. Items key:value pairs for data set RECFM=F or V.....	887
462. X-IBM-Attributes=base and data set RECFM=U.....	888
463. Items key:value pairs for data set RECFM=U.....	888
464. Contents of the create a sequential and partitioned data set document.....	889
465. Contents of the JSON file list document.....	890
466. Contents of the unix file and directory.....	891
467. Unix items key:value pairs.....	891
468. Contents of the Mount and unmount a file system document.....	892
469. Contents of the create a UNIX file document.....	892
470. Contents of the JSON error report document.....	893
471. Error categories for z/OS data set and file REST interface operations.....	894
472. Category 1 errors for z/OS data set and file REST interface operations.....	894
473. Category 2 errors for z/OS data set and file REST interface operations.....	896

474. Category 3 errors for z/OS data set and file REST interface operations.....	897
475. Category 4 errors for z/OS data set and file REST interface operations.....	898
476. Category 5 errors for z/OS data set and file REST interface operations.....	898
477. Category 6 errors for z/OS data set and file REST interface operations.....	898
478. Category 7 JSON parser conditions.....	900
479. Category 8 z/OS XL C/C++ Conditions.....	900
480. Category 9 errors for z/OS data set and file REST interface operations.....	900
481. Operations provided through the z/OS jobs REST interface services.....	900
482. Job modify operations provided through the z/OS jobs REST interface services.....	903
483. JESJOBS class authorizations needed for performing job modify operations.....	904
484. Variations of a PUT request for submitting a job from a data set or UNIX file.....	922
485. Contents of the JSON job document.....	936
486. Contents of the JSON job notification document.....	938
487. Contents of the JSON job feedback document.....	940
488. Contents of the JSON job file document.....	940
489. Contents of the JSON job step data document.....	942
490. Contents of the JSON error report document.....	942
491. Error categories for z/OS jobs REST interface operations.....	943
492. Category 1 errors.....	944
493. Category 3 errors.....	945
494. Category 4 errors.....	945
495. Category 5 errors.....	945
496. Category 6 errors.....	946
497. Category 7 errors.....	951
498. Category 8 errors.....	951

499. Category 9 errors.....	952
500. Category 10 errors.....	953
501. z/OS Management Services Catalog services: operations summary.....	953
502. Response from a z/OS Management Services Catalog request failure.....	956
503. Query parameters.....	956
504. Array of objects.....	958
505. Fields for a catalog service object.....	963
506. Fields for each catalog service input definition object (csInputDefinition).....	966
507. Fields for the csInputValueMetadata object.....	969
508. z/OS Management Services Catalog default widget selection for the string inputs.....	969
509. Fields for each catalog service input logic object (csInputLogic).....	970
510. Fields for each dependent logic object (csDependentLogic).....	970
511. Fields for each catalog service page object (csPage).....	971
512. Fields for Catalog Service Input Definition Validation (csInputValidation).....	971
513. Fields for Catalog Service RunAsUser Step (runAsUserSteps).....	972
514. Fields for Catalog Services RunAsUser Step Approvals (stepApprovals).....	973
515. Fields for an ActivityTimeLine object.....	973
516. Fields for a Category object	976
517. Query parameters.....	979
518. Array of objects.....	981
519. Fields for a service submission create object.....	983
520. Fields for a get service submission object	984
521. Fields for an ActivityTimeLine object.....	986
522. Fields for a get service submission object.....	991
523. Fields for an ActivityTimeLine object.....	993

524. Fields for modifying a service submission object.....	996
525. Fields that may be required to be provided before a draft service submission can be successfully submitted.....	998
526. Fields for adding a comment to a service submission object.....	998
527. Response from a get list of job statements request.....	1000
528. Response from a get list of target systems request.....	1001
529. Operations provided through the z/OSMF settings services.....	1006
530. z/OSMF authentication services method.....	1013
531. Response body for a "log in to the z/OSMF server" request.....	1014
532. HTTP error response codes for a "log in to the z/OSMF server" request.....	1015
533. Request content for a change user password or passphrase request.....	1016
534. Response body for a change user password or passphrase request.....	1017
535. HTTP error response codes for a change user password or passphrase request.....	1017
536. z/OSMF workflow services: operations summary.....	1020
537. z/OSMF workflow services: URI path variables.....	1021
538. Error response body elements for the z/OSMF workflow services API.....	1021
539. Request content for the create workflow request.....	1024
540. HTTP error response codes for a create workflow request.....	1030
541. Response from a create workflow request.....	1030
542. HTTP error response codes for a get workflow properties request.....	1034
543. JSON object that is returned to a get workflow properties request.....	1034
544. Get Workflow Properties request: Format of the automation-info object.....	1038
545. Get Workflow Properties request: Format of the step-info object.....	1040
546. Get Workflow Properties request: Format of the jobInfo object.....	1048
547. Get Workflow Properties request: Format of the jobstatus object.....	1048
548. Get Workflow Properties request: Format of the jobfiles object.....	1049

549. Get Workflow Properties request: Format of the variable-reference object.....	1050
550. Get Workflow Properties request: Format of the variable-info object.....	1050
551. List workflows request: Format of the workflow-info object.....	1055
552. Request content for the start workflow request.....	1058
553. HTTP error response codes for a start workflow request.....	1059
554. Structure of the JSON object that is returned to the notification URL.....	1060
555. HTTP error response codes for a cancel workflow properties request.....	1062
556. HTTP error response codes for a delete workflow request.....	1064
557. HTTP error response codes for a retrieve a workflow definition request.....	1066
558. JSON object that is returned to a retrieve a workflow definition request.....	1066
559. Retrieve a workflow definition request: Fields included in every step-definition object.....	1068
560. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow).....	1068
561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step.....	1069
562. Retrieve a workflow definition request: Format of the variable-specification-info object.....	1073
563. Retrieve a workflow definition request: Format of the variable-definition object.....	1074
564. HTTP error response codes for an archive workflow request.....	1079
565. Response from an archive workflow request.....	1079
566. HTTP error response codes for a get archived workflow properties request.....	1081
567. List archived workflows request: Format of the workflow-info object.....	1081
568. HTTP error response codes for a get archived workflow properties request.....	1083
569. JSON object that is returned to a get archived workflow properties request.....	1083
570. Get Archived Workflow Properties request: Format of the automation-info object.....	1087
571. Get Archived Workflow Properties request: Format of the step-info object.....	1088
572. Get Archived Workflow Properties request: Format of the variable-reference object.....	1095
573. Get Archived Workflow Properties request: Format of the variable-info object.....	1095

574. HTTP error response codes for a delete archived workflow request.....	1097
575. Summary of template step elements.....	1120
576. Elements for saving a generated job.....	1124
577. Summary of REST step elements.....	1132
578. Use scope and isCallable to coordinate workflow-to-workflow actions.....	1138
579. Major step elements for defining a called workflow in your workflow definition file.....	1139
580. Information to specify for notifying a user about a suspended step.....	1146
581. How a runAsUser ID is used in workflow processing.....	1150
582. Basic usage scenarios: How static or dynamic determination is made for the runAsUser value....	1153
583. Translatable strings.....	1154
584. Internal variables: Variables that are provided with z/OSMF.....	1164
585. Reference tables for the Workflows XML schema.....	1173
586. Workflow metadata elements	1175
587. Sub-elements for a configuration type workflow.....	1179
588. Sub-elements for a provisioning type workflow.....	1181
589. Workflow upgrade elements	1183
590. Manifest elements summary table.....	1185
591. Step elements summary table: Elements to use for defining all steps.....	1187
592. Step elements summary table: Additional sub-element for parent steps.....	1189
593. Step elements summary table: Additional sub-elements for leaf steps.....	1191
594. Step elements summary table: Additional sub-elements for REST steps.....	1206
595. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow).....	1210
596. Variable definition elements summary.....	1214
597. Variable definition type-specific sub-elements summary.....	1217
598. atCreate element summary.....	1221

599. Summary of benefits for each integration type.....	1223
600. Functions with data sets and Unix files	1227
601. Parameters of editorWithContentfunction.....	1228
602. Parameters of editorWithPath function.....	1228
603. Parameters of utilPathSelector function.....	1229
604. Parameters of editorCompare function.....	1230
605. Parameters of editorCompareWithContent function.....	1231
606. Parameters of Download function.....	1231
607. Parameters of the Upload function.....	1232
608. Function usage with z/OSMF window.....	1234
609. Functions you can add to the zosmfExternalTools API.....	1234
610. Functions provided in the AppLinker API.....	1241
611. Expected results by launching option and task state.....	1247
612. Methods provided for the client side logger.....	1252
613. URL to use for each z/OSMF category.....	1265

About this document

This document is intended to help you write programs that use the services and facilities of IBM® z/OS Management Facility (z/OSMF).

The programming interfaces in z/OSMF include the following.

- Representational State Transfer (REST) services for working with z/OS and z/OSMF.
- XML schema for creating a workflow definition for performing activities on a z/OS system, such as configuring a component or product.
- Services for creating plug-ins that add installation-specific function to z/OSMF.

Who should use this document

This information is intended for the programmer responsible for writing programs that use the z/OSMF infrastructure. Such activities, include, for example:

- Using an application programming interface (API) as a client to interact with z/OS programmatically .
- Developing a JavaScript program that includes a graphical user interface (GUI).

This document assumes that you are familiar with the z/OS operating system and its accompanying products.

Where to find more information

For an overview of the information associated with z/OS, see *z/OS Information Roadmap*.

IBM Z and LinuxOne Community

To access a rich community of business and technical expert blogs and forums for z/OSMF, visit the IBM Z and LinuxOne Community at <https://ibm.github.io/zOSMF/>.

z/OSMF One Stop Hub

For the latest developer news about z/OSMF, visit the z/OSMF One Stop Hub at [z/OSMF One Stop Hub \(ibm.github.io/zOSMF/\)](https://ibm.github.io/zOSMF/).

z/OSMF Community Guild

To join z/OSMF subject matter experts in deep dives of technical demos and learn about improvements to the platform, visit the z/OSMF Community Guild at <https://ibm.biz/zosmfguildhome>.

z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to z/OS concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time that it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your web browser to the following website, which is available to all users (no login required): <https://www.ibm.com/docs/en/zos-basic-skills?topic=zosbasics/com.ibm.zos.zbasics/homepage.htm>.

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead [“If you have a technical problem”](#) on page lxi.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function

If your comment or question is about z/OS itself, submit a request through the <https://ibm-z-hardware-and-operating-systems.ideas.ibm.com/>.

Feedback on IBM Documentation function

If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content

If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: IBM z/OSMF Programming Guide, SC27-8420-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the [IBM Support Portal \(support.ibm.com\)](https://support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy \(www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument\)](http://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OSMF Programming Guide for Version 2 Release 5 (V2R5)

The following content is new, changed, or no longer included in V2R5. The most recent updates are listed at the beginning of each section.

New information

The following content is new.

December 2022 refresh

- APAR PH44884 added new REST APIs to Storage Management to add volumes to a SMS storage group and perform Source Control Data Set (SCDS) modification, validation and activation. For more information, see [“Storage management services”](#) on page 547.
- APAR PH43962 added the capability for workflow users to have their runAsUser workflow step signed so that the steps can be automatically executed under their credentials during workflow automation. For more information, see [“runAsUser identity for a step”](#) on page 1149 and [“Workflow XML reference”](#) on page 1173.
- APAR PH43962 added new sub-element *maintainFragments* to the workflow information element. For more information, see [“Specifying the workflow metadata”](#) on page 1109.

October 2022 refresh

- APAR PH44152 added new parameter *X-IBM-Notification-Options* to the z/OSMF jobs REST interface. For more information, see [“Submit a job”](#) on page 918

September 2022 refresh

- APAR PH45201 added a new REST API for Software Management. For more information, see [“Deleting the Temporary Catalog Aliases”](#) on page 502.
- APAR PH47050 added new REST APIs for Software Management. For more information, see [“Missing FIXCAT Updates”](#) on page 536, [“Missing Critical Updates”](#) on page 531, and [“Software Update Search”](#) on page 540.

June 2022 refresh

- Resource pool services have been updated to support NJE and SMS for LPAR provisioning. The following APIs have new fields added:
 - [“Create an LPAR resource pool entry”](#) on page 78
 - [“Modify an LPAR resource pool entry”](#) on page 82
 - [“Retrieve a list of LPAR resource pool entries”](#) on page 87
 - [“Retrieve the properties of an LPAR resource pool entry”](#) on page 91
 - [“Obtain an LPAR resource pool entry”](#) on page 95

- Software services instances services have been updated to display the version of the Software Services template that an instance was provisioned from. The following APIs have the template-version field added:
 - [“Create a software services instance” on page 321](#)
 - [“Get the contents of a software services instance” on page 334](#)
 - [“List the software services instances” on page 345](#)
- Resource management services have been updated to support archive history entries. For more information, see [“Get a domain” on page 104](#).
- Added a new API section for List Template Resource Pools and updated the table in the Resource management services section. For more information, see [“List template resource pools” on page 186](#) and [“Resource management services” on page 102](#).
- Made updates to the [“z/OS data set and file REST interface” on page 813 APIs](#).
- z/OSMF Core Multi-sysplex added two new APIs that allows the user to check the connection of multiple z/OSMF systems.
 - [“Check all MVS systems in the local plex” on page 709](#).
 - [“Validate the status of the selected system” on page 703](#).

May 2022 refresh

- APAR PH37308 adds the z/OS compliance REST interface, which is used in the collection of z/OS compliance data. For more information, see the [“z/OS Compliance REST Interface” on page 760](#).

February 2022 refresh

- The Security Configuration Assistant (SCA) REST services are new in this release. For more information, see [“Security Configuration Assistant services” on page 446](#) (APAR PH41248).
- Added new parameter, *workflowArchiveSAFID*, to the z/OSMF workflow REST services. See [“Create a workflow” on page 1023](#) and [“Get the properties of an archived workflow” on page 1082](#).

January 2022 refresh

- APAR PH41196 created a new API section for Get system variables and updated the table in the z/OS system variable services section. For more information, see [“z/OS system variable services” on page 802](#) and [“Get system variables” on page 805](#).

December 2021 refresh

- z/OS Management Services Catalog task has a new application with a new set of REST APIs. For more information, see [“z/OS Management Services Catalog services” on page 953](#).
- The storage management services are enhanced with the addition of the following REST services:
 - [“Get a list of storage classes” on page 614](#)
 - [“Get a storage class definition” on page 621](#)
 - [“Get a list of data classes” on page 580](#)
 - [“Get a data class definition” on page 597](#)
- The sysplex management services are new in this release. For more information, see [“Sysplex management services” on page 658](#). The following CFRM policy editor REST services are added:
 - [“Get a list of CFRM policies” on page 659](#)
 - [“Get CFRM policy properties” on page 661](#)
 - [“Activate CFRM policy” on page 667](#)
 - [“Rename existing CFRM policy” on page 668](#)
 - [“Modify CFRM policy properties” on page 670](#)
 - [“Add CFRM policies to administrative policy” on page 676](#)
 - [“Copy existing CFRM policy with a new name” on page 684](#)

- [“Delete multiple CFRM policies from an administrative policy” on page 685](#)

September 2021 GA

- The resource pool services are enhanced with the addition of the following REST services, which are used for working with LPAR resource pools:
 - [“Create an LPAR resource pool entry” on page 78](#)
 - [“Modify an LPAR resource pool entry” on page 82](#)
 - [“Delete an LPAR resource pool entry” on page 86](#)
 - [“Retrieve a list of LPAR resource pool entries” on page 87](#)
 - [“Retrieve the properties of an LPAR resource pool entry” on page 91](#)
 - [“Obtain an LPAR resource pool entry” on page 95](#)
 - [“Release an LPAR resource pool entry” on page 100](#)
- The following resource management services are updated in support of LPAR resource pools:
 - [“Get a resource pool” on page 158](#)
 - [“List the resource pools” on page 173](#)
 - [“Get a domain resource pool” on page 164](#)
 - [“List domain resource pools” on page 179](#)
- The storage management services are new in this release. These REST interfaces allow programs to retrieve information about storage groups and volumes on the z/OS system. For more information, see [“Storage management services” on page 547](#). The following REST services are added:
 - [“Get a list of storage groups” on page 549](#)
 - [“Get a storage group definition” on page 560](#)
 - [“Get a list of volumes” on page 571](#)
 - [“Get a volume definition” on page 576](#)
- The software management services are enhanced with the addition of the following REST services:
 - [“Deploy a software instance” on page 509](#)
 - [“Delete a Deployment” on page 519](#)
- Note was added regarding browser limitations for large response object sizes in the Swagger UI. For more information, see [Chapter 1, “Using the z/OSMF REST services,” on page 1](#).

Summary of changes for z/OSMF Programming Guide for Version 2 Release 4 (V2R4)

The following content is new, changed, or no longer included in V2R4. The most recent updates are listed at the beginning of each section.

New

The following content is new.

December 2021 refresh

- New deployment REST APIs were added. For more information, see [“Deploy a software instance” on page 509](#) and [“Delete a Deployment” on page 519](#).
- z/OS Management Services Catalog task has a new set of REST APIs. For more information, see [“z/OS Management Services Catalog services” on page 953](#).

June 2021 refresh

- The following z/OS console service was added: [“Get messages from a hardcopy log” on page 795](#).

- The following TSO/E address space services were added:
 - [“Get the response to a command that was issued with the TSO/E REST API” on page 734](#)
 - [“Issue a TSO/E command with z/OSMF REST API” on page 725](#)
- Usage notes were added to [“TSO/E address space services” on page 714](#).
- For the service described in [“Start a workflow” on page 1056](#), the JSON object that is returned to the notification URL now includes the user ID of the user who stopped automation.

May 2021 refresh

- New Software Management workflow variables are added. For more information, see [Appendix D, “Software Management workflow variables,” on page 1297](#).

February 2021 refresh

The z/OS jobs REST interface services are enhanced, as follows:

- Customer header X-IBM-Target-System-User is new. This header is used to supply a z/OS user ID for accessing the target system.
- Customer header X-IBM-Target-System-Password is new. This header is used to supply the password that is associated with the z/OS user ID specified on the X-IBM-Target-System-User header.

For more information, see [“z/OS data set and file REST interface” on page 813](#) and [“z/OS jobs REST interface” on page 900](#).

December 2020 refresh

New function is available for IBM Cloud Provisioning and Management when you install the PTF for APAR PH29813.

The Cloud Provisioning REST services are enhanced in the following ways:

- A resource pool defines the scope of shared z/OS resources within a cloud domain that has multiple tenants. In this release, the concept of a shared resource pool is expanded to include sharing resources across an entire domain. Here, the resource pool is referred to as a *domain-shared resource pool*. Previously, you were limited to sharing a resource pool within a single tenant. By allowing multiple tenants within a domain to share a resource pool, you simplify the management of resources in a cloud provisioning environment. In contrast, if resource isolation is wanted for a template, it is recommended that you define a tenant-specific shared resource pool or a dedicated resource pool. The Resource Management services are enhanced with new properties to create, view, and modify domain-shared resource pools. The Software Services services are updated to allow a domain-shared resource pool to be associated with a template. The names of shared resource pools end with two wildcard qualifiers (*.*).

The following REST services are new in this release:

- [“Get a domain history” on page 110](#)
- [“Get a tenant history” on page 126](#)
- [“Get a domain resource pool” on page 164](#)
- [“Get a resource pool history” on page 171](#)
- [“List domain resource pools” on page 179](#)
- [“Get a software services template history” on page 244](#)
- [“Get a published software service template history” on page 302](#)
- [“Modify a published software service template” on page 315](#)
- The following Cloud Provisioning REST services have new or enhanced functions in this release:
 - [“Get a domain” on page 104](#)
 - [“List the domains” on page 112](#)
 - [“Get a resource pool” on page 158](#)
 - [“Create a software services template” on page 203](#)

- [“Create a new version of a software services template ” on page 211](#)
- [“Modify a software services template” on page 221](#)
- [“Get a software services template” on page 228](#)
- [“Test a software services template” on page 261](#)
- [“Run a published software service template” on page 285](#)
- [“Get a published software service template” on page 290](#)
- [“Create a software services instance” on page 321](#)
- [“Get the contents of a software services instance” on page 334](#)
- [“List the software services instances” on page 345](#)
- [“Update a software services instance” on page 359](#)
- [“Get the response for an action performed against a software services instance” on page 380](#)
- [“List the responses for actions performed against a software services instance” on page 385](#)
- When you create a template, you can specify new options to do the following:
 - Delete instances automatically when they are deprovisioned. Previously, it was always necessary to delete a deprovisioned instance manually.
 - Archive provisioning workflows automatically when they complete. Previously, you were limited to selecting to either keep or delete a workflow. Archiving is the new default behavior for the workflows disposition property.
- The cloud provisioning actions schema file is updated to support a description element for each action. With the description element, you can optionally provide a text description of the action and the function it performs. For more information, see [Chapter 4, “Preparing software to exploit cloud provisioning,” on page 1277](#).
- The actions editor is enhanced to work with variables of all types. Previously, the editor was limited to working with string variables only.
- It is now possible to provide an optional description for the provisioning actions that you define. A new "description" field is returned in the action structure for the following REST services:
 - [“Create a software services instance” on page 321](#)
 - [“Update a software services instance” on page 359](#)

Common HTTP request header **X-IBM-Session-Limit-Wait** is added to the z/OS data set and file REST interface services. You can include this header to extend the amount of time that a request can take to find an available CEA TSO address space. For more information, see [“z/OS data set and file REST interface” on page 813](#).

Prior to the December refresh

- APAR PH24527 adds new REST APIs, which can be used to enable and disable z/OSMF plug-ins and services. For more information, see [“z/OSMF settings services” on page 1006](#).
- In support of the new ability to manipulate portable software instance objects, the following REST APIs are added to software management services:
 - [“List the portable software instances defined to z/OSMF” on page 507](#)
 - [“Retrieve the properties of a portable software instance” on page 520](#)
 - [“Add a new portable software instance” on page 525](#)
 - [“Delete a portable software instance” on page 530](#)
- The following topic is new: [“System variables” on page 1165](#).
- The z/OS console services REST interface now includes properties that you can use to control console attributes, such as 'auth', 'routcode', 'mscope', 'storage', and 'auto'. For more information, see [“Issue a command from a system console” on page 765](#).
- In Cloud Provisioning, you can now provision composite templates and clustered composite templates in a multiple sysplex domain. Previously, this capability was available for standard templates only.

- The z/OS jobs REST interface services are enhanced, as follows:
 - Custom header **X-IBM-Target-System** is added to the z/OS jobs REST interface services. You can include this header to direct a REST request to a remote system in your enterprise. For more information, see [“z/OS jobs REST interface” on page 900](#).
 - Custom header **X-IBM-Intrdr-File-Encoding** is new. You can include this header to specify the appropriate code page for data that is written to the internal reader. For more information, see [“Submit a job” on page 918](#).
 - With the addition of the exec - data query parameter, the following z/OS jobs REST interface services can optionally return execution data about jobs on z/OS:
 - [“Obtain the status of a job” on page 905](#)
 - [“List the jobs for an owner, prefix, or job ID” on page 909](#)
 - With the addition of the status query parameter, the following GET request can be limited to active jobs only: [“List the jobs for an owner, prefix, or job ID” on page 909](#).
 - With the addition of the **fileEncoding** query parameter, the following GET request can specify an alternative code page for the spool file on z/OS: [“Retrieve the contents of a job spool file” on page 914](#).
- The z/OS data set and file REST interface services are enhanced, as follows:
 - The following options are added to the customer header X-IBM-Data-Type:
 - Value `"text;fileEncoding=<codepage>"` can be used to select an alternative EBCDIC code page. The default code page is IBM-1047.
 - Value `text;crLf=true` can be used to control whether each input text line is ended with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.
 - Customer header Content-Encoding is new. This header is used to compress the response data. If present, its value indicates which encoding method can be used to decompress the media-type that was specified in the Accept-Encoding header. The supported file format is gzip.

For more information, see [“z/OS data set and file REST interface” on page 813](#).

- In Cloud Provisioning, you can define a multiple sysplex domain, with which you can provision middleware across multiple sysplexes in your enterprise. In this configuration, creating and modifying objects is done from a primary z/OSMF system, from which you can provision templates on other, secondary z/OSMF systems. This enhancement allows your cloud provisioning environment to scale beyond the scope of a single sysplex.

In support of multiple sysplex domains, the following Cloud Provisioning services REST APIs are added:

- [“Create a tenant” on page 117](#)
- [“Delete a tenant” on page 135](#)
- [“Assign CPU properties to a tenant” on page 136](#)
- [“Assign memory capping properties to a tenant” on page 138](#)
- [“Assign a solution ID” on page 140](#)
- [“Disable CPU capping” on page 141](#)
- [“Disable memory capping” on page 143](#)
- [“Disable metering” on page 144](#)
- [“Enable CPU capping” on page 146](#)
- [“Enable memory capping” on page 147](#)
- [“Enable metering” on page 149](#)
- [“Add tenant consumer” on page 150](#)
- [“Remove tenant consumer” on page 152](#)
- [“Add tenant description” on page 153](#)

- [“Add tenant groups” on page 155](#)
- [“Remove tenant groups” on page 156](#)
- [“Update the security state for a tenant” on page 192](#)
- [“Get data set attributes” on page 76](#)
- New REST APIs that can be used to obtain and delete web tokens for authentication with an application server. For more information, see [“z/OSMF authentication services” on page 1013](#).
- The z/OSMF Workflows schema is enhanced. z/OSMF makes all of the publicly visible instance variables for the calling workflow available to the called workflow. These variables are referred to as *caller scope* variables. They are shared only with the called workflow. For more information, see [“Caller scope variables” on page 1169](#).
- A new property for workflows was added to the software management REST API. For more information, see [“Software management services” on page 458](#) and [Appendix C, “Understanding the Portable Software Instance descriptor file,” on page 1289](#).
- A new property for data set property labels was added to the software management REST API. For more information, see [“Software management services” on page 458](#) and [Appendix C, “Understanding the Portable Software Instance descriptor file,” on page 1289](#).
- A z/OSMF workflow includes a number of “built-in” variables that workflow authors can use. These variables are called *workflow internal variables*. You can reference them in your workflow definition without the need for you to define them. For more information, see [“Workflow internal variables ” on page 1163](#).
- A new field that is called `requires-zcx-addr` is added to [“Obtain an IP address” on page 52](#).

Changed

The following content is changed.

April 2021 refresh

- The HTTP error status response codes are changed. For more information, see [Table 532 on page 1015](#)

January 2021 refresh

- The `izud-datasets` workflow variable was updated to include the `izud-cataloged` and `izud-catalog` properties. For more information, see [Appendix D, “Software Management workflow variables,” on page 1297](#).

December 2020 refresh

The description for z/OS data set and file REST interface services error reporting category 8 is updated with information about dynamic allocation errors. For more information, see [“Category 8 — z/OS XL C/C++ Conditions” on page 900](#).

Prior to the August 2020 refresh

- The description for the `volume-serial` field was corrected in [“Add a new software instance” on page 476](#).
- The software management services Export REST API was updated to allow the input of a work data set name prefix and volume or storage class. For more information, see [“Export a defined software instance” on page 482](#).
- Global variables are deprecated. Use instance variables or system variables instead. Future releases might not support global variables.

Deleted

The following content was deleted.

December 2020 refresh

- Mention of return of a stack trace for error handling was removed from the Software Management REST APIs section. For more information, see [“Software management services” on page 458](#).

Prior to the July 2020 refresh

- The tables that list all HTTP status reason codes were removed from the software management services section. The reason codes themselves, which are returned to the calling client, include all needed information. For more information, see [“Software management services” on page 458](#).

Information applicable to all releases

This information contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line in the margin by the change.

The *Readers' Comments - We'd Like to Hear from You* section at the end of this publication has been replaced with a new section [“How to send your comments to IBM” on page lxi](#). The hardcopy mail-in form has been replaced with a page that provides information appropriate for submitting comments to IBM.

Chapter 1. Using the z/OSMF REST services

z/OSMF supports the use of Representational State Transfer (REST) APIs, which are public APIs that your application can use to work with system resources and extract system data. As with implementations of REST services on other platforms, the z/OSMF APIs allow for easy-to-use services that are language- and platform-independent, stateless, scalable, and easily parsed.

Processing overview

The z/OSMF REST services can be invoked by any hypertext transfer protocol (HTTP) client application, running on the z/OS local system or a remote system. The services support requests in either of the following protocols: HTTP/1.0 or HTTP/1.1.

Conceptually, your application (the client) issues requests to the target system (z/OS) in the form of request messages. The product name is z/OS. Each request message consists of a request line, optionally followed by request headers (HTTP headers), an empty line, and an optional message body. The request line includes the HTTP method, such as GET, a Universal Resource Locator (URL) and, where appropriate, parameters that further qualify the request. The URL is required and must be URI-encoded as specified in RFC 2396. For more information about RFC 2396, see the [Uniform Resource Identifiers \(URI\): Generic Syntax \(www.ietf.org/rfc/rfc2396.txt\)](http://www.ietf.org/rfc/rfc2396.txt) web page.

If the API determines that the request is valid, it performs the requested service. After the API performs the service, it creates an HTTP response. If the request is successful, this response takes the form of an HTTP 200 (OK) response and, if applicable, an object that contains a result set. Depending on which service was requested, the result set might be returned in a format that requires parsing by your program, for example, a JavaScript Object Notation (JSON) object. In other cases, results might be returned in another format, such as plain text or binary data. If the request is not successful, the response consists of a non-OK HTTP response code with details of the error that is provided in the form of a JSON object.

It is assumed that users of these services are familiar with the JSON standard and coding practices. The following references provide more helpful information:

- Hypertext Transfer Protocol 1.1: <http://www.w3.org/Protocols/>
- Multipurpose Internet Mail Extensions (MIME) media types: <http://www.iana.org/assignments/media-types/index.html>
- Introducing JSON: <http://www.json.org>.

Using the Swagger interface

You can use the Swagger interface to display information about the following z/OSMF REST APIs:

- [“Cloud provisioning services” on page 46](#)
- [“Notification services” on page 438](#)
- [“TSO/E address space services” on page 714](#)
- [“z/OS console services” on page 763](#)
- [“z/OS data set and file REST interface” on page 813](#)
- [“z/OS jobs REST interface” on page 900](#)
- [“z/OSMF information retrieval service” on page 1002](#)
- [“z/OSMF workflow services” on page 1020](#)
- [“Storage management services” on page 547](#)
- [“Sysplex management services” on page 658](#)

To enable the use of Swagger at your installation, you must define the Swagger resources in your external security manager, and grant READ access to the appropriate users and groups. Ask your security administrator to do the following:

- You and z/OSMF administrators require READ access to the IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers resource in the EJBROLE class. The following example shows RACF® commands that grant the access:

```
RDEFINE EJBROLE IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers
UACC(NONE)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE)
ID(IZUUSER) ACCESS(READ)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE)
ID(IZUADMIN) ACCESS(READ)
```

- z/OSMF administrators require READ access to the IZUDFLT.com.ibm.ws.management.security.resource.Administrator resource in the EJBROLE class. The following example shows RACF commands that grant the access:

```
RDEFINE EJBROLE IZUDFLT.com.ibm.ws.management.security.resource.Administrator UACC(NONE)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE)
ID(IZUADMIN) ACCESS(READ)
```

After the access is defined, use this address in a web browser to display information about the REST APIs:

```
https://<hostname>:<port>/zosmf/ibm/api/explorer
```

Supply the appropriate value for hostname and port.

Note: Because of limitations in some web browsers, large response objects might not be rendered in their entirety in the Swagger UI. If so, consider using request filters to reduce the amount of returned data or try another web browser.

The Swagger Document allows you to try z/OSMF REST APIs without the need to write the program. Alternatively, you can also use the command line tool like CURL to access z/OSMF REST APIs without having to write the program. The following examples demonstrate how to write CURL commands to access z/OSMF REST APIs:

Note: In the following examples:

```
Dataset-name: <dataset name>
z/OSMF username: <username>
z/OSMF user password: <password>
The hostname of z/OSMF server: < hostname >
The port of z/OSMF server: <port>
The body of the request in json format file: inputdata.json
The body of the request in text format file: inputdata.txt
```

The following example shows a sample CURL command to issue a POST request. This request can be used to create a sequential or partitioned data set.

```
curl -X POST -k -u "<username>:<password>" -d"inputdata.json" --header "X-CSRF-ZOSMF-HEADER: dummy" --header "Accept: application/json" --header "Content-Type: application/json" "https://<hostname>:<port>/zosmf/restfiles/ds/<dataset name>"
```

Note: In the sample command for the Json format input data, header "Content-Type: application/json" should be included.

The following example shows a sample CURL command to issue a GET request. This request can be used to retrieve data from a sequential data set.

```
curl -X GET -k -u "<username>:<password>" --header "X-CSRF-ZOSMF-HEADER: dummy" --header "Accept: application/json" --header "Content-Type: text/plain" "https://<hostname>:<port>/zosmf/restfiles/ds/<dataset name>"
```

Note: In this sample command, the output of this API is text format, header "Content-Type: text/plain" should be included.

The following example shows a sample CURL command to issue a PUT request. This request can be used to write data to an existing sequential data set.

```
curl -X put -k -u "<username>:<password>" -d@"inputdata.txt" --header "X-CSRF-ZOSMF-HEADER: dummy" --header "Accept: application/json" --header 'Content-Type: text/plain' "https://<hostname>:<port>/zosmf/restfiles/ds/<dataset name>"
```

Note: In this sample command, for the text format input data, header "Content-Type: text/plain" should be included.

The following is a sample CURL command to issue a DELETE request. This request can be used to delete sequential and partitioned data sets on a z/OS system.

```
curl -X DELETE -k -u "<username>:<password>" --header "X-CSRF-ZOSMF-HEADER: dummy" --header "Accept: application/json" "https://<hostname>:<port>/zosmf/restfiles/ds/<dataset name>"
```

Authenticating to z/OSMF

The z/OSMF REST services API is a secure interface that requires authentication. z/OSMF supports the following methods for passing user credentials to the API:

- Basic authentication
- Certificate authentication
- Web token authentication

Basic authentication means that the client program provides a z/OS user ID and password in the header of the initial request.

The following topics describe each method of authentication:

- [“Basic authentication” on page 3](#)
- [“Certificate authentication” on page 4](#)
- [“Web token authentication” on page 4](#)

Basic authentication

Your client program authenticates to the z/OSMF REST services API with a valid z/OS user ID and password.

Typically, authentication is done through the HTTP header that is included in each client request. However, you might also consider using a single sign-on technique, such as the following:

1. On the first request to the server, the client request includes a basic HTTP authentication header that contains a valid user ID and password. The header property value pair should look like this example:

```
'Authorization': 'Basic <encoding of userid:password>'
```

where <encoding of userid:password> is a base 64 encoding of <userid>:<password>.

2. On successful log-in, your application receives the following values in the response header:
 - HTTP status code 200
 - Lightweight Third Party Access (LTPA) token, which contains the credentials for your program. For z/OSMF, the token is a LtpaToken2 value, which supports strong encryption.
3. On subsequent requests, your program supplies the LTPA token for authentication with the z/OSMF REST services API, instead of the basic HTTP authorization header. You can provide the LTPA token through the Cookie header property, for example:

You can provide the LTPA token through the Cookie header property, for example:

```
'Cookie': '<ltpaToken2=<tokenvalue>'
```

followed by the token, for example:

```
'Cookie': 'LtpaToken2=IExabotu2sfNbJij6rajHJcFiDi  
1H0hm13yKvylwfJ4q8goCFEYH41FQN1AgdMMVP6/nVbH/IKw  
015b7ZqWuZ8nd0YcECAJg1ss2Vq4q21C1jLvVGTyNLk6ivbgs  
7oQWM98bSuAN1Qtvlrx9uZ8EY4GqqscaErQ09vzAhwgkcedWB  
jn21LNj1+G8o1JA4uB+Cv5XamrUvziY2pcbCKjFjNt5EQ97Nf2  
sBzv1anfrENhV9u0LRpw9DibzKLh0R1f0rp5xySAe7Ery69  
eynt4ItaVWCcpt+CYHFbpHpW/C7INWHeNcaNktz0DBmHh6EW1; '
```

Certificate authentication

Your client program authenticates to the z/OSMF REST services API with a certificate.

With a client certificate, your program can access z/OSMF without having to supply a user ID and password. If your client is browser-based, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

To use client certificate authentication for the z/OS jobs REST interface services, you:

1. Create the certificate. You can create the certificate in RACF or import it into RACF.
2. Distribute the certificate to the appropriate workstations, for example, FTP it in binary.
3. Import the certificate into the browser, by using the procedure that is appropriate for that browser.

For more information about creating and managing digital certificates, see [RACF and digital certificates in z/OS Security Server RACF Security Administrator's Guide](#). For an example, see [Scenario 5: Creating client browser certificates with a locally signed certificate in z/OS Security Server RACF Security Administrator's Guide](#).

Web token authentication

The z/OSMF authentication services API can be used to obtain authentication tokens (a JSON Web Token and an LTPA token) on the user's authentication request. This API can also be used to delete the current store of JSON Web Tokens and LTPA tokens. This API is provided for z/OSMF tasks and vendor applications. For more information, see [“z/OSMF authentication services” on page 1013](#).

Web token support must be enabled on your z/OSMF system. For more information, see [Enabling JSON Web Token support in IBM z/OS Management Facility Configuration Guide](#).

Supported HTTP versions

The z/OSMF REST interface services support requests in either of the following protocols: HTTP/1.0 or HTTP/1.1

Usage considerations for the z/OSMF REST services

Observe the following considerations when you use the z/OSMF REST interfaces:

- As with any z/OSMF task or function, the REST services compete for z/OSMF resources with users of the z/OSMF web browser interface. Thus, concurrent high usage of the REST services can affect response time for users of the z/OSMF web browser interface.
- During periods of concurrent high usage of the REST services, an application can experience connection failures, such as connection refused, connection timed out, or connection reset. In these cases, the application should try the request again. The number of retry attempts needed depends on how much work is being requested of the server. It might be necessary for your installation to modify the workload and reduce the arrival rate of requests.
- Some browsing environments do not support all of the HTTP methods, such as HTML 4 or XHTML 1, or might block applications from accessing response content that has a non-successful HTTP response

status code (4nn and 5nn). As a workaround, your application can use the following custom HTTP request headers:

X-IBM-Requested-Method:

GET, PUT, and DELETE requests can be "tunneled" through a **POST** method by using this custom HTTP header.

X-IBM-Bypass-Status:

If set to true, all response status codes are set to 200, and the custom HTTP response header **X-IBM-Actual-Status** is included in the returned data. To determine the original status code, your application must check the **X-IBM-Actual-Status** header.

Timestamp data type

The timestamp data type is used in the definition of some data models and notification message formats in the z/OSMF REST services APIs. Where it appears in this information, the timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.

Enabling browser log in through a client certificate

It is possible to run the z/OS jobs REST interface services directly from a web browser. Here, you must first authenticate to z/OSMF through your browser. In z/OSMF, authentication is typically done by entering your user ID and password at the **Welcome** page. However, it is also possible to log in with a client certificate, if your installation favors this approach. With a client certificate, you can access z/OSMF through your browser without having to supply a user ID and password.

When a client certificate is created, it requires security associations. For example, on a system with RACF installed, your client certificate can be associated with a RACF user ID. After the user ID is associated with the client certificate, the certificate can be exported. You must get the client certificate from the server side by using the FTP command and typing in the hostname or IP address of the server. When prompted, enter your user ID and password, and use the **bin** command to transfer the file in binary format. Then, import the client certificate into the client's browser.

In client certificate authentication, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

If z/OSMF does not accept your client certificate, z/OSMF displays the Welcome page for you to enter your user ID and password.

If your installation plans to enable client certificate login for the z/OS jobs REST interface services, understand that it is your responsibility to create the certificate and manage its distribution to users. It is recommended that you ensure that users have browsers that support importing a certificate.

For more information about creating digital certificates, see [RACF and digital certificates](#) in *z/OS Security Server RACF Security Administrator's Guide*.

Allowing cross-site access to REST services

The z/OSMF REST services can be accessed by an HTTP client application, or by a web application that is running on the same host system. By default, z/OSMF blocks access attempts from web applications on other host systems. In such cases, the request is failed with error message IZUG846W, which indicates that a Cross Site Request Forgery (CSRF) was attempted. In security terminology, a *CSRF* is a type of malicious attack on a website in which the attacker sends unauthorized commands to a web server or web application from a user that the server or application trusts.

You can enable your applications to make cross-site z/OSMF REST requests. To do so, you must perform the following steps:

1. Review your applications and identify those applications that use the z/OSMF REST services.
2. For any applications that make cross-site requests to z/OSMF REST services, update the application by adding the following HTTP custom header to every cross-site request:

X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or an empty string ("").

3. Because most modern web browsers block cross-site access due to same origin restrictions, an extra step is required for browser applications — you must define the origin site of the web application to your installation security "allow list." Instructions for doing so are provided in [“Enabling cross-origin resource sharing \(CORS\) for REST services”](#) on page 6.

Non-browser applications, such as Java™ applications, require only the custom header. They do not require an "allow list" definition.

Note: z/OSMF includes the parmlib member IZUPRMxx, which your installation can use to override the z/OSMF default settings. IZUPRMxx includes the setting `CSRF_SWITCH=ON | OFF` to allow for disabling CSRF checking for all requests to the z/OSMF server. By default, `CSRF_SWITCH` is set to ON to ensure that your installation is protected against CSRF attacks. However, in some limited cases, such as for testing, you might choose to temporarily disable CSRF checking by setting `CSRF_SWITCH=OFF`. However, it is recommended that you leave this setting enabled to prevent CSRF attacks.

Origin header

The Origin header indicates the origin of the cross-site access request (that is, the server that initiated the request). If you include the Origin header in a cross-site request, include the internet protocol (HTTP or HTTPS) with the origin domain. Otherwise, your request is blocked by the CSRF filter, which is enabled by default (`CSRF_SWITCH=ON`).

The following example shows the use of the Origin header with a z/OSMF REST API request. Notice that the internet protocol HTTP is included in the Origin header (in bold text).

```
PUT /zosmf/restconsoles/consoles/P083598 HTTP/1.1
Host: sys123.yourco.com
Content-Type:application/json
X-CSRF-ZOSMF-HEADER:zosmf
Origin:https://sys456.yourco.com
{
  "cmd": "F AXR,JFPAXR D T",
  "system": "sys123"
}
```

Enabling cross-origin resource sharing (CORS) for REST services

Your installation can allow browser applications from certain, trusted sites to access z/OSMF REST services on the host system. If so, you must enable cross-origin resource sharing (CORS) on the host system. This work involves creating an "allow list" of exceptions (the trusted sites), and enabling those exceptions in your external security manager.

Identify which sites are to be allowed, and which REST interfaces are to be made available for cross-site access. Then, work with your security administrator to create the appropriate authorizations in your external security manager. In a RACF installation, for example, define generic or discrete profiles for the remote sites in the ZMFAPLA class, and permit the profiles to the z/OSMF REST interfaces.

To define a profile for a remote site, use the following format:

```
<SAF_PREFIX>.REST.<identifier>.<reversed-hostname>
```

Where:

- `<SAF_PREFIX>` is the SAF prefix for your z/OSMF configuration. By default, the prefix is IZUDFLT.

- REST.<identifier> identifies the REST interface that is to be allowed for use by the remote site. Table 1 on page 7 shows the identifiers for each of the z/OSMF interfaces. To indicate all REST interfaces, specify an asterisk (*) as the identifier.

Table 1. SAF identifiers for the z/OSMF REST interfaces

REST interface	Identifier
Application Linking Manager interface services	APPLINK
Application server routing services	GATEWAY
Cloud provisioning services	PROVISIONING
Data persistence services	PERSIST
Multisystem routing services	GATEWAY
MVS™ subsystem services	FILES
Security Configuration Assistant services	CONFIGURATION
Software management services	SWMGMT
Storage management services	STORAGE
Sysplex management services	SYSPLEX
Topology services	SYSTEM
TSO/E address space services	TSO
z/OS console REST interface services	CONSOLE
z/OS data set and file REST interface services	FILES
z/OS jobs REST interface services	JOBS
z/OSMF workflow services	WORKFLOW

- <reversed-hostname> is the site's fully qualified domain name in reverse, for example, the domain WWW.IBM.COM would be specified as COM.IBM.WWW. Use uppercase letters for any alphabetic characters in the profile. Specify a domain name only; not the full URL. Omit the protocol (HTTP:// or HTTPS://).

If the site is known by an IP address, specify the IP address in reverse. For any IP addresses that you define, it is recommended that you create a discrete profile (without wildcards) for each address. Use a valid "dotted decimal" Internet Protocol version 4 (IPv4) address. IPv6 addresses and internationalized domain name (IDN) addresses are not supported.

Though not recommended, you can use generic profiles with wildcard characters to allow access from multiple domains or subdomains. You might do so temporarily to allow for internal testing of the REST interfaces across multiple sandbox systems.

For example, assume that your installation wants to allow a web browser application on the site "lab2.ibm.com" to send requests for z/OS jobs REST interface services on the site "lab1.ibm.com." To allow a web browser application to issue cross-site requests from the "lab2" site, your RACF security administrator would create the authorization on the "lab1.ibm.com site," as follows:

1. Create a profile for the "lab2.ibm.com" site in the ZMFAPLA class. In the profile, include the identifier to represent the resource (the REST interface), for example, JOBS to indicate the z/OS jobs REST interface. In RACF, enter the **RDEFINE** command, as follows:

```
RDEFINE ZMFAPLA IZUDFLT.REST.JOBS.COM.IBM.LAB2 UACC(NONE)
```

To allow this cross-site access for all of the REST interfaces, specify a wildcard (*) in place of a specific resource identifier. For example,:

```
RDEFINE ZMFAPLA IZUDFLT.REST.*.COM.IBM.LAB2 UACC(NONE)
```

2. Ensure that the z/OSMF server user ID (by default, IZUSVR) has READ access to the profile. In RACF, enter the **PERMIT** command, as follows:

```
PERMIT IZUDFLT.REST.JOBS.COM.IBM.LAB2 CLASS(ZMFAPLA) ID(IZUSVR) ACCESS(READ)
```

3. Refresh the ZMFAPLA class. In RACF, enter the **SETROPTS** command, as follows:

```
SETROPTS RACLIST(ZMFAPLA) REFRESH
```

Note: These settings do not override the security settings in your browser. For more information about cross-origin resource sharing (CORS), see the documentation for the web browser.

Application Linking Manager interface services

To perform traditional system management tasks in z/OS, you might interact with several different interfaces, such as the TSO command line, graphical user interfaces, and web-style interfaces. With the z/OSMF Application Linking Manager, it is possible to link or connect some of these tasks and external applications together for a smoother user experience.

The key components in the Application Linking Manager process include the:

- **Event requestor.** z/OSMF task or external application that requests the launch of a specific function within another task or external application
- **Event.** Action requested by the event requestor. It includes the type of event and the event parameters.
- **Event type.** Object that connects an event requestor to an event handler. It identifies the handlers that can process an event and the possible parameters that can be supplied with an event.
- **Event handler.** z/OSMF task or external application that can process the event parameters and display the requested information.

Figure 1 on page 8 depicts the relationship of these components in the application linking process. For more information about these components and to obtain a list of the predefined event types, requestors, and handlers, see [“Event types, requestors, and handlers shipped with z/OSMF”](#) on page 10.

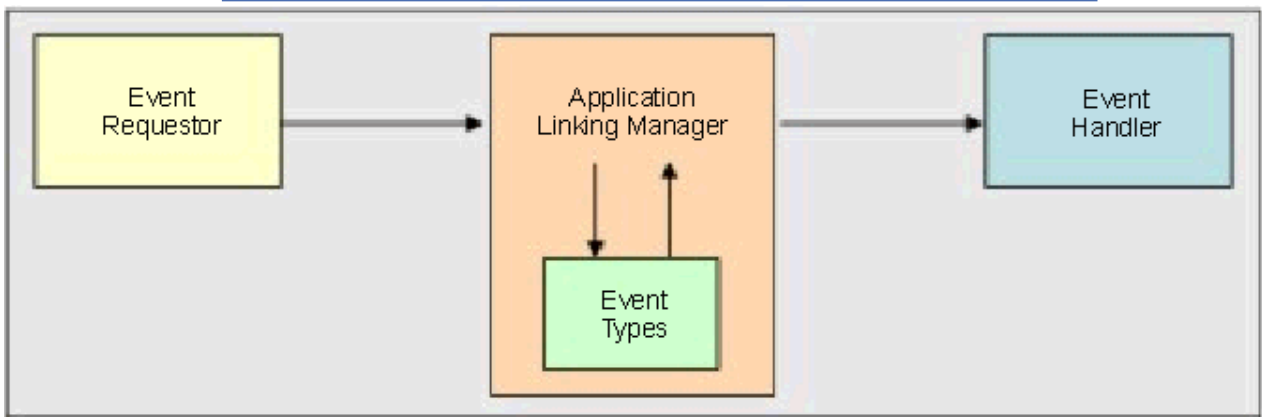


Figure 1. Key components in the application linking process

z/OSMF provides the following resources for working with the Application Linking Manager:

- **Application Linking Manager task**, which provides a graphical user interface that you can use to add, query, or remove event type and event handler definitions.
- **Application Linking Manager REST APIs**, which are a set of REST services that allows a client application to add, query, or remove event type and event handler definitions.
- **AppLinker JavaScript API**, which is a set of JavaScript services that allows a client application to send events to the Application Linking Manager or to define the context to be displayed. The JavaScript services are applicable only if you are creating your own z/OSMF plug-in.

The remainder of this section describes the Application Linking Manager REST APIs. For information about the Application Linking Manager task, see the z/OSMF online help. For details about the AppLinker JavaScript API, see [“Using the Application Linking Manager JavaScript APIs” on page 1240](#).

Operations provided through the Application Linking Manager interface services

Table 2 on page 9 lists the operations that the Application Linking Manager interface services provide.

Table 2. Operations provided through the Application Linking Manager interface services	
Operation	HTTP method and URI path
“Register an event type” on page 24	POST /zosmf/izual/rest/eventtype
“Register an event handler” on page 26	POST /zosmf/izual/rest/handler?eventId=<eventId>
“Obtain a list of all tasks that are eligible to be handlers” on page 29	GET /zosmf/izual/rest/adm/getHandlerEligibleTasks?eventId=<eventId>
“Obtain a list of handlers for an event type” on page 30	GET /zosmf/izual/rest/handler?eventId=<eventId>
“Unregister an event handler” on page 31	DELETE /zosmf/izual/rest/handler/<handlerId>?eventId=<eventId>
“Unregister an event type” on page 32	DELETE /zosmf/izual/rest/eventtype/<eventId>

Required authorizations

To submit requests through the Application Linking Manager interface services, your user ID requires authorization to the Application Linking Manager task. Ensure that your user ID has READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.ADMINTASKS.APPLINKING. By default, any user ID with z/OSMF administrator authority can access the Application Linking Manager interface services.

Further, if you plan to use the Application Linking Manager interface services to list the registered event handlers for an event type, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <SAF_PREFIX> in the APPL class.
- READ access to the <SAF_PREFIX>.*.izuUsers profile in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for sent data and returned data; for the detailed format of each returned object, see the JSON object description for each operation.

Error handling

For errors that occur during the processing of a request, the Application Linking Manager interface returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the problem or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad Request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to the Application Linking Manager task.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the Application Linking Manager interface services are logged in the z/OSMF runtime log files or the z/OSMF server logs directory. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Invoking a z/OSMF application link externally

It is possible for an external application (a web application, for example, or a desktop application), to launch a z/OSMF event on a z/OS system. Here, the application issues a command comprised of the URL for the local instance of z/OSMF, combined with the appropriate event information.

The following is an example of such a request. In the example, *eventType-1* identifies the event type and *value-1* specifies the parameter input for the event handler.

```
https://...:9443/zosmf/?izual.eventType={eventType-1}&parm1={value-1}
```

This command launches z/OSMF in a new browser window on the issuer's system, and sends the event to z/OSMF after session startup. If the user is authenticated to z/OSMF, the handler is launched in the user's session. Otherwise, the user is prompted to authenticate before the handler can be launched.

Event types, requestors, and handlers shipped with z/OSMF

IBM ships several event types, requestors, and handlers with z/OSMF so that you can quickly start exploiting the application linking capability and easily navigate between multiple z/OSMF tasks.

For more details about these application linking objects, see the following sections:

- [“Event types” on page 11](#)
- [“Event requestors” on page 15](#)
- [“Event handlers” on page 19](#)

Event types

An *event type* is the intermediary that connects an event requestor to an event handler. Event requestors and event handlers do not interact directly; instead, z/OSMF uses the event type to pass information, the event type and parameters, from the requestor to the handler.

Event handlers can process all, none, or a subset of the parameters that are provided with an event type. When creating an event requestor for an event type, ensure that the event requestor supplies the parameters that are required by the event handlers. To obtain a list of the parameters that are required for the IBM-supplied handlers to process an event type, see [“Event handlers” on page 19](#).

For a list of the event types that are shipped with z/OSMF, see [Table 3 on page 12](#). This table provides the following information for each event type: ID, display name, description, parameters, and the name of the plug-in that registers the event type. By default, these event types are listed in the Application Linking Manager task if the corresponding plug-in is configured in your z/OSMF instance.

Note: This table is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Table 3. Event types shipped with z/OSMF

Event Type ID	Display Name	Description	Parameters Provided	Registered By
IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW	Create a workflow	Perform a guided set of steps, for example, to configure components or products in your installation.	<p>workflow_definition_file_name Fully-qualified UNIX file name or PDS name of the workflow definition file.</p> <p>workflow_name Name that the user has provided for the workflow.</p> <p>workflow_owner z/OS user id of the workflow owner.</p> <p>workflow_target_system Name of the system for this workflow.</p>	Workflows plug-in
IBM.ZOSMF.WORKFLOWS.EDIT_WORKFLOW	Edit a Workflow Definition	Launches the workflow editor for a specified workflow definition file when context is provided. Otherwise, a dialog is displayed to prompt the user to select a workflow definition file for editing.	<p>workflow_definition_file_name Fully-qualified UNIX file name of the workflow definition file.</p> <p>variable_input_file_name Fully qualified UNIX file name of the workflow variable input properties file.</p> <p>target_step_name Name of a step in the workflow definition file to be displayed on initial launch of the workflow editor.</p> <p>launch_read_only A value of "true" indicates the workflow editor is launched in read-only mode. Modifications to the workflow definition and related files are not allowed.</p>	Workflow plug-in
IBM.ZOSMF.CONFIGURE_NETWORK_POLICIES	Configure network policies	Configure z/OS Communications Server network policies.	No parameters are provided for this event type.	Network Configuration Assistant plug-in
IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION	View Active Service Definition	View the active service definition.	<p>sysplex Name of the sysplex for which to display the active service definition.</p> <p>timestamp Timestamp in milliseconds when the service definition was active.</p>	Workload Management plug-in

Table 3. Event types shipped with z/OSMF (continued)

Event Type ID	Display Name	Description	Parameters Provided	Registered By
IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS	View Report Class of Active Service Definition	View the report classes that are contained in the active service definition.	sysplex Name of the sysplex for which to display the active service definition. timestamp Timestamp in milliseconds when the service definition was active. reportClass Name of the report class to be viewed.	Workload Management plug-in
IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS	View Service Class of Active Service Definition	View the service classes that are contained in the active service definition.	sysplex Name of the sysplex for which to display the active service definition. timestamp Timestamp in milliseconds when the service definition was active. serviceClass Name of the service class to be viewed. period Period in the service class to be viewed.	Workload Management plug-in
IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD	View Workload of Active Service Definition	View the workloads that are contained in the active service definition.	sysplex Name of the sysplex for which to display the active service definition. timestamp Timestamp in milliseconds when the service definition was active. workload Name of the workload to be viewed.	Workload Management plug-in
IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY	View Active Service Policy	View the active service policy.	sysplex Name of the sysplex for which to display the active service policy. timestamp Timestamp in milliseconds when the service policy was active.	Workload Management plug-in
IBM.ZOSMF.VIEW_DATASET	View Data Set	View or browse a data set.	dataSetName Name of the dataset to be viewed.	ISPF plug-in

Table 3. Event types shipped with z/OSMF (continued)

Event Type ID	Display Name	Description	Parameters Provided	Registered By
IBM.ZOSMF.VIEW_JOB_STATUS	View Job Status	View the status of a job.	jobName Name of the job for which to display status. jobId ID of the job for which to display status.	ISPF plug-in
IBM.ZOSMF.VIEW_SYSPLEX_PERF	View Sysplex Performance	View the overall performance of a sysplex.	sysplex Name of the sysplex for which to display the performance.	Resource Monitoring plug-in
IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX	View Performance Index Details	View the performance index of a sysplex.	sysplex Name of the sysplex for which to display the performance index.	Resource Monitoring plug-in
IBM.ZOSMF.VIEW_WLM_REPORT_CLASS_PERF	View Report Class Performance	View the execution velocity and response time metrics for the report classes in a sysplex.	sysplex Name of the sysplex for which to display performance metrics. reportClass Name of the report class for which to display metrics.	Resource Monitoring plug-in
IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF	View Service Class Performance	View the performance of active service class periods in a sysplex.	sysplex Name of the sysplex for which to display performance metrics. serviceClass Name of the service class for which to display metrics. period Period in the service class for which to display metrics.	Resource Monitoring plug-in
IBM.ZOSMF.VIEW_WLM_STATUS	View WLM Status	Display the status of WLM in the sysplex.	sysplex Name of the sysplex for which to display the status of WLM.	Workload Management plug-in
IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF	View Workload Performance	View the execution velocity and response time metrics for the workloads in a sysplex.	sysplex Name of the sysplex for which to display performance metrics. workload Name of the workload for which to display metrics.	Resource Monitoring plug-in

Event requestors

An *event requestor* provides a user-interface control that when invoked passes an event type and the required parameters to the Application Linking Manager task so that they can be passed to an appropriate handler. [Table 4 on page 16](#) lists the name and description of the event requestors that are shipped with z/OSMF, and it provides a list of the event types the requestor can invoke, the user-interface (UI) control you can use to invoke the event type, and the IBM-supplied handlers for the event type.

When creating an event handler for the IBM-supplied event types, you can use the information provided in [Table 4 on page 16](#) to invoke the event type and verify that your handler is displaying the expected output.

Notes:

1. The UI controls listed in [Table 4 on page 16](#) are available only if the corresponding event type is registered in z/OSMF, and one or more handlers are registered for the event type and are available to process the request. Otherwise, the UI control is disabled or is not provided in the user interface.
2. [Table 4 on page 16](#) is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Table 4. Event requestors shipped with z/OSMF				
Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
Application Linking Manager task	Create context-sensitive launch points between z/OSMF tasks or external applications.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	<ul style="list-style-type: none"> Import action provided in the Event Types table. Import action provided in the Handlers table. 	Import Manager task
Incident Log task	Diagnose system problems, and send diagnostic data to IBM or other vendors for further diagnostics.	IBM.ZOSMF.VIEW_DATASET	<ul style="list-style-type: none"> View Log button on the Diagnostic Data tab on the View Diagnostic Details page. Link in the Source Name column in the Diagnostic Data table on the Diagnostic Data tab. 	ISPF task
Incident Log task	Diagnose system problems, and send diagnostic data to IBM or other vendors for further diagnostics.	IBM.ZOSMF.VIEW_JOB_STATUS	View Job Details action provided in the FTP Job Status table on the FTP Job Status page.	By default, no handler is provided for this event type.
Links task	Add links to external sites for system management tools and information.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	New action provided in the Links page.	Import Manager task
Resource Monitoring task	Monitor the performance of the z/OS, AIX®, Linux®, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS	View WLM Service Class action provided in the bar chart for metrics that are organized or filtered by WLM service class or WLM service class period.	Workload Management task
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD	View WLM Workload action provided in the bar chart for metrics that are organized or filtered by WLM workload.	Workload Management task
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS	View WLM Report Class action provided in the bar chart for metrics that are organized or filtered by WLM report class or WLM report class period.	Workload Management task
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	System Status link provided in the Select Sysplex window.	System Status task
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX	<ul style="list-style-type: none"> View > Performance Index Details action provided in the Resources table on the System Status page. Link in the Performance Index Status column in the Resources table on the System Status page. 	Resource Monitoring task

Table 4. Event requestors shipped with z/OSMF (continued)

Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION	<ul style="list-style-type: none"> • View > Active WLM Service Definition action provided in the Resources table on the System Status page. • Link in the Related Service Definition column in the Resources table on the System Status page. 	Workload Management task
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY	<ul style="list-style-type: none"> • View > Active WLM Policy action provided in the Resources table on the System Status page. • Link in the Active WLM Policy column in the Resources table on the System Status page. 	Workload Management task
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_WLM_STATUS	View > WLM Status action provided in the Resources table on the System Status page.	Workload Management task
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	View performance of systems link provided on the WLM Status page.	System Status task
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF	<ul style="list-style-type: none"> • View performance of active policy link provided on the WLM Status page. • View Performance of Active Policy action provided in the table on the Service Policies for Sysplex page. • View Performance of Active Policy action provided in the Service Policies table in a View or Modify tab. • View Performance of Selected and View Performance of All actions provided in the Service Classes and Service Class Overrides tables in a View or Modify tab. 	Resource Monitoring task
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF	View Performance of Selected and View Performance of All actions provided in the Workloads table in a View or Modify tab.	Resource Monitoring task

Table 4. Event requestors shipped with z/OSMF (continued)				
Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMFVIEW_WLM_REPORT_CLASS_PERF	View Performance of Selected and View Performance of All actions provided in the Report Classes table in a View or Modify tab.	Resource Monitoring task

Event handlers

An *event handler* is a z/OSMF task or external application that can handle requests sent by event requestors. Event handlers support specific event types and can support all, none, or a subset of the parameters that are supplied with an event type. For a handler to process a request, it must receive the correct parameters from the event requestor.

For a list of the event types and parameters (optional and required) that are supported by the handlers shipped with z/OSMF, see [Table 5 on page 21](#). This table also provides a description of each handler and the expected output for each event type and handler combination.

Note: This table is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Table 5. Event handlers shipped with z/OSMF					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
Network Configuration Assistant task	Configure TCP/IP policy-based networking functions.	IBM.ZOSMF.CONFIGURE_NETWORK_POLICIES	None	None	Displays the main page in the Network Configuration Assistant task.
Import Manager task	Add installation-specific function to z/OSMF in the form of plug-ins.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	tab	None	Opens to the Import Manager task.
ISPF task	Access traditional ISPF applications.	IBM.ZOSMF.VIEW_DATASET	dataSetName	None	Displays the data set.
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. <ul style="list-style-type: none">Important service class periodsAll service class periodsReport class periods	Opens a dashboard that contains the performance index for the following items: <ul style="list-style-type: none">Important service class periodsAll service class periodsReport class periods
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_REPORT_CLASS_PERF	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. reportClass If unspecified, the metrics for all report classes are displayed.	Opens a dashboard that contains the following metrics for each WLM report class: <ul style="list-style-type: none">Execution velocityResponse time
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. serviceClass If unspecified, the metrics for all service classes are displayed. period If unspecified, the metrics for all periods in the specified service class are displayed. This value is used only if a service class is specified.	Opens a dashboard that contains the following metrics for each WLM service class period: <ul style="list-style-type: none">Performance indexExecution velocityExecution velocity goalResponse timeResponse time goal
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. workload If unspecified, the metrics for all workloads are displayed.	Opens a dashboard that contains the following metrics for each WLM workload: <ul style="list-style-type: none">Execution velocityResponse time
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	None	sysplex If unspecified, the default value is the z/OSMF host sysplex.	Displays a list of the sysplexes that are defined in the System Status task.

Table 5. Event handlers shipped with z/OSMF (continued)					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
Workflows task	Perform a guided set of steps, for example, to configure components or products in your installation.	IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW	workflow_definition_file_name Fully-qualified UNIX file name or PDS name of the workflow definition file. workflow_name Name that the user has provided for the workflow. workflow_owner z/OS user id of the workflow owner. workflow_target_system Name of the system for this workflow.	workflow_callback Callback URL to use when the workflow is created. For example, the caller can be notified with the name of workflow, to be used for a subsequent invocation to launch the workflow. workflow_comments Comments to be added to the workflow on creation.	Opens to the <i>Workflow Details</i> panel for the newly created workflow in the Workflows task.
z/OSMF Workflow Editor Handler	Provide a set of tools for viewing and editing a workflow definition.	IBM.ZOSMF.WORKFLOWS.EDIT_WORKFLOW	None	workflow_definition_file_name Fully-qualified UNIX file name of the workflow definition file. variable_input_file_name Fully qualified UNIX file name of the workflow variable input properties file. target_step_name Name of a step in the workflow definition file to be displayed on initial launch of the workflow editor. launch_read_only A value of "true" indicates the workflow editor is launched in read-only mode. Modifications to the workflow definition and related files are not allowed.	Opens to the workflow editor main page for a specified workflow definition file, or a dialog that prompts the user to select a workflow definition file for editing.
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. timestamp If unspecified, the service definition that is currently active is displayed.	Displays the service definition that is active in the sysplex.
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS	None	sysplex If unspecified, the default value is the z/OSMF host sysplex. timestamp If unspecified, the service definition that is currently active is displayed. reportClass If unspecified, all the report classes in the service definition are displayed.	Displays the specified report class or a list of all the report classes defined in the active service definition.

Table 5. Event handlers shipped with z/OSMF (continued)					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS	None	<p>sysplex If unspecified, the default value is the z/OSMF host sysplex.</p> <p>timestamp If unspecified, the service definition that is currently active is displayed.</p> <p>serviceClass If unspecified, all the service classes in the service definition are displayed.</p> <p>period If unspecified, all the periods in the service class are displayed. This value is used only if a service class is specified.</p>	Displays the specified service class and period, or a list of all the service classes defined in the active service definition.
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD	None	<p>sysplex If unspecified, the default value is the z/OSMF host sysplex.</p> <p>timestamp If unspecified, the service definition that is currently active is displayed.</p> <p>workload If unspecified, all the workloads in the service definition are displayed.</p>	Displays the specified workload or a list of all the workloads defined in the active service definition.
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY	None	<p>sysplex If unspecified, the default value is the z/OSMF host sysplex.</p> <p>timestamp If unspecified, the service policy that is currently active is displayed.</p>	Displays the service policy that is active in the sysplex.
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_STATUS	None	<p>sysplex If unspecified, the default value is the z/OSMF host sysplex.</p>	Opens the WLM Status tab.

Register an event type

You can use this operation to define a new event type to z/OSMF.

HTTP method and URI path

```
POST /zosmf/izual/rest/eventtype
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **eventtype** identifies the event type component of the application linking process.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include a JSON object that describes the event type to be registered, for example:

```
{
  id: "IBM.ZOSMF.EVENT_TYPE_ID",
  displayName: "Default English name",
  desc: "Default English description",
  owner: "ownerId",
  params: {
    "key1": "English description of the parameter.",
    "key2": "English description of the parameter."
  }
}
```

Figure 2. Registering an event type: request content

The following values are supported:

id

Specify a unique identifier for the event type. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (_). The event ID is required and must be unique.

It is recommended that IDs have the format *company-name.product-name.event-name* where:

- *company-name* is the name of your company. Use a period as the delimiter within the company name. For example, for Tivoli® products, the company name can be *IBM.TIVOLI*.
- *product-name* is the name of the product for which the event type is being created.
- *event-name* is the action that will be completed by the event handler. The event name should start with a verb that reflects this action. Use an underscore as the delimiter within the event name.

For example, to create an event type that allows a user to view the status of a job listed in z/OSMF, the event name portion of the ID can be *VIEW_JOB_STATUS*. The entire ID can be *IBM.ZOSMF.VIEW_JOB_STATUS*.

displayName

Specify the name of the event type. The name is required and can contain up to 50 characters.

Example: *View job status.*

desc

Specify a description of the event type. The description is optional and can contain up to 200 characters.

Example: *Use this event type to view the status of a job. This event type is invoked when a user selects the View Job Status action.*

owner

Specify the ID of the first z/OSMF task or external application that registered the event type. Typically, event types are registered by or on behalf of event handlers. They can also be registered by or on behalf of event requestors. This field is required, and can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (_).

params

Specify the name and description of each parameter that event requestors can supply with an event. Enter each parameter name and description combination on a separate line, and enclose the entry in quotes. Use a colon to separate the parameter name and description, and a comma to end each entry. The final entry is not ended with a delimiter.

This area can contain up to 4,000 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), underscores (_), and commas (,).

For example, to allow event requestors to provide a job ID, job name, and user ID for an event type that displays the status of a job, you would specify the following parameters:

```
params: {  
  "jobID": "ID assigned to the job.",  
  "jobName": "Name specified for the job.",  
  "userID": "ID of the user who submitted the job."  
}
```

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Required authorizations

See [“Required authorizations” on page 9.](#)

IBM-supplied event types

z/OSMF includes a number of predefined event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF” on page 10.](#)

Example of registering an event type

A sample request to register an event type is shown in [Figure 3 on page 25.](#)

```
POST /zosmf/izual/rest/eventtype HTTP/1.1  
Host: zosmf1.yourco.com  
  
Accept: application/json  
Content-Type: application/json  
  
{  
  "id": "IBM.ZOSMF.VIEW_JOB_STATUS",  
  "displayName": "View Job Status",  
  "owner": "SDSF",  
  "params": { "jobName": "Name of the job for which to view status." }  
}
```

Figure 3. Example: Registering an event type

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 10](#).

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error and result fields, as shown in [Figure 4 on page 26](#).

```
{"error":null,"result":null}
```

Figure 4. Example: Returned results of a successful event registration

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtxt. The example in [Figure 5 on page 26](#) shows the results for an attempt to register an already-registered event with different parameters:

```
{
  "error": {
    "msgid": "IZUG690E",
    "msgtxt": "Event type \"IBM.ZOSMF.VIEW_JOB_STATUS\" is already defined, but
              different parameters are specified."},
  "result": null
}
```

Figure 5. Example: Returned results of an unsuccessful event registration

Register an event handler

You can use this operation to define a new event handler to z/OSMF.

HTTP method and URI path

```
POST /zosmf/izual/rest/handler?eventTypeId=<eventTypeId>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **eventTypeId=<eventTypeId>** is the event type to be associated with the new handler.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include a JSON object that describes the event handler to be registered.

The following values are supported:

type

Handler type. For a z/OSMF plug-in, specify INTERNAL. For an external application, specify EXTERNAL.

id

Unique identifier for a launch point within the handler task or application. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (_). The handler ID is required and must be unique.

For external applications, you can specify any value. To ensure uniqueness, it is recommended that you make the company name and the application name part of the handler ID. For example, *IBM.TIVOLI.OMEGAMON*.

For applications within the ISPF task, it is recommended that you prefix the handler ID with IBM.ISPF. Then, specify an ID for the application that will handle the events. For example, *IBM.ISPF.SDSF.ST*.

applId

Identifier assigned to the z/OSMF plug-in that contains the task. It is required for a z/OSMF task (type is set to INTERNAL). Omit this value if the handler is an external application.

Table 6 on page 27 lists the valid applId values for the z/OSMF tasks.

Table 6. Valid applId values for the z/OSMF plug-ins	
z/OSMF Task	applId Value
Network Configuration Assistant task	CAV1R11
Import Manager task	IzuImportManager
ISPF task	com.ibm.zosmf.ispf
Resource Monitoring task	IZUR
System Status task	IZUR
Workflows task	workflow
Workload Management task	IZUW

taskId

Identifier assigned to the z/OSMF task. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (_). The task ID is required when type is set to INTERNAL. Omit this value if the handler is an external application.

Table 7 on page 27 lists the valid task ID values for the z/OSMF tasks.

Table 7. Valid taskId values for the z/OSMF tasks	
z/OSMF Task	taskId Value
Network Configuration Assistant task	Configuration Assistant
Import Manager task	IZUG_TASK_zOSMFImportManager
ISPF task	ISPF
Resource Monitoring task	IZUR_PERFDESKS_TASK_ID
System Status task	IZUR_OVERVIEW_TASK_ID
Workflows task	Workflows
Workload Management task	Workload Management

displayName

For the handler name, specify the name of the handler task or application. The name is required and can contain up to 50 characters. For z/OSMF tasks, it is recommended that you use the same name displayed in the z/OSMF desktop. For example, *Workload Management*.

For external applications, it is recommended that you use the name of the product or application. For example, *Omegamon*.

url

URL to be used for accessing the handler. The URL can contain up to 4,000 characters, including alphanumeric characters (A-Z, a-z, 0-9), blanks, mathematical symbols (+ - = | ~ () { } \), punctuation marks (? , . ! ; : ' " / []), and the following special characters: %, \$, #, @, ^, *, and _. The URL is required and must be URI-encoded as specified in RFC 2396. For more information about RFC 2396, see the [Uniform Resource Identifiers \(URI\): Generic Syntax \(www.ietf.org/rfc/rfc2396.txt\)](http://www.ietf.org/rfc/rfc2396.txt) web page.

For a z/OSMF task, specify a URL that is relative to the z/OSMF instance. That is, the URL must begin with `/zosmf/`. For an external application, specify the full URL, including the protocol.

options

The `CONTEXT_SUPPORT` option indicates what the handler will display when it processes events of this type. Specify one of the following values for `CONTEXT_SUPPORT`:

OPT_CONTEXT_SUPPORT_NONE

Handler is launched without context. That is, its homepage is displayed. If the handler is already open, it receives focus, but the context is not updated.

If the handler is an external application, it opens in a separate window. Otherwise, the handler opens in a new z/OSMF task tab.

This option is selected by default.

OPT_CONTEXT_SUPPORT_LAUNCH

Handler is launched with context. If the handler is already open, it receives focus, but the context is not updated.

If the handler is an external application, it opens in a separate window. Otherwise, the handler opens in a new z/OSMF task tab.

OPT_CONTEXT_SUPPORT_LAUNCH_AND_RELOAD

Handler is launched with context. If the handler is already open, a message is displayed warning the user that the current context will be overwritten. This option is supported only when the event requestor and handler are z/OSMF tasks.

OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH

Handler is launched with the context it specified when subscribing to the event type. This option is supported only when the event requestor and handler are z/OSMF tasks.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 9](#).

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 10](#).

IBM-supplied event handler registrations

z/OSMF includes a number of predefined event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF” on page 10](#).

Obtain a list of all tasks that are eligible to be handlers

The *handlerEligible* property indicates whether a z/OSMF task can participate in the application linking process as an event handler. To obtain a list of the tasks with the *handlerEligible* property set to *true*, use the GET method.

HTTP method and URI path

```
GET /zosmf/izual/rest/adm/getHandlerEligibleTasks?eventId=<eventId>
```

where:

- **/zosmf/izual/rest/adm** identifies the Application Linking Manager interface.
- **getHandlerEligibleTasks** indicates that the service will retrieve a list of tasks that are eligible to be event handlers.
- **eventId=<eventId>** is the event type for which the request is being submitted.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 9](#).

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 10](#).

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error field, and the result field lists the task ID, navigation URL, display name, and plug-in ID for each task that is eligible to be a handler. [Figure 6 on page 30](#) provides a sample response for a successful request.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"error":null,"result":{"Task":[
{"TaskID":"test2","navigationUrl":"/zosmf/test2","displayName":"test2","PluginID":"TestPlugin1"},
{"TaskID":"test3","navigationUrl":"/zosmf/test3","displayName":"test3","PluginID":"TestPlugin1"},
{"TaskID":"test4","navigationUrl":"/zosmf/test4","displayName":"test4","PluginID":"TestPlugin1"},
{"TaskID":"test1","navigationUrl":"/zosmf/test1","displayName":"Test1","PluginID":"TestPlugin2"}
]}}
```

Figure 6. Sample response from a successful list tasks request

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtext.

Obtain a list of handlers for an event type

You can use this operation to obtain a list of registered handlers for an event type.

HTTP method and URI path

```
GET /zosmf/izual/rest/handler?eventType=<eventType>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **eventType=<eventType>** is the ID of the event type for which you want to obtain a list of registered handlers.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 9.

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling”](#) on page 10.

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error field, as shown in [Figure 7 on page 31](#), [Figure 8 on page 31](#), and [Figure 9 on page 31](#).

If the request is successful and one or more handlers are enabled for the event type, information about the registered handlers are returned, as depicted in [Figure 7 on page 31](#).

```
{
  "result": [
    {
      "id": "IBM.ZOSMF.IZU_IMPORT_HANDLER",
      "taskId": "IZUG_TASK_zOSMFImportManager",
      "enabled": true,
      "defaultHandler": false,
      "applId": "IzuImportManager",
      "type": "INTERNAL",
      "displayName": "Import Manager",
      "url": "\/zosmf\/IzuImportUtility\/index.jsp",
      "eventType": "IBM.ZOSMF.IMPORT_EXTERNAL_APP",
      "options": {
        "CONTEXT_SUPPORT": "OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"
      }
    }
  ],
  "error": null
}
```

Figure 7. Example: Handlers enabled for the event type

If the request is successful and no handlers are registered for the event type, the result field contains null data, as depicted in [Figure 8 on page 31](#).

```
{
  "result": null,
  "error": null
}
```

Figure 8. Example: Returned results of a successful list handlers request

If the request is successful and all the handlers that are registered for the event type are disabled, the result field contains an empty array, as depicted in [Figure 9 on page 31](#).

```
//Result if all the handlers defined for the event type are disabled.
{
  "result": [],
  "error": null
}
```

Figure 9. Example: Returned results of a successful list handlers request

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtext.

Unregister an event handler

You can use this operation to remove an existing event handler registration from z/OSMF.

HTTP method and URI path

```
DELETE /zosmf/izual/rest/handler/<handlerId>?eventType=<eventType>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **<handlerId>** is the ID of the event handler to be removed.
- **eventType=<eventType>** is the ID of the event type for which you want to remove the specified handler. The combination of **<handlerId>** and **<eventType>** identifies the handler registration to be removed.

Standard headers

None.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Required authorizations

See [“Required authorizations” on page 9.](#)

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 10.](#)

Unregister an event type

You can use this operation to remove the definition of an event type from z/OSMF.

HTTP method and URI path

```
DELETE /zosmf/izual/rest/eventtype/<eventTypeId>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **eventtype** identifies the event type component of the application linking process.
- **<eventTypeId>** is the ID of the event type to be removed.

Standard headers

None.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Required authorizations

See [“Required authorizations” on page 9.](#)

Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 10](#).

Application server routing services

The application server routing services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. Use these services to route requests and responses between the client-side and server-side code for any z/OSMF plug-ins you created where the server-side code is hosted on an application server other than the z/OSMF server.

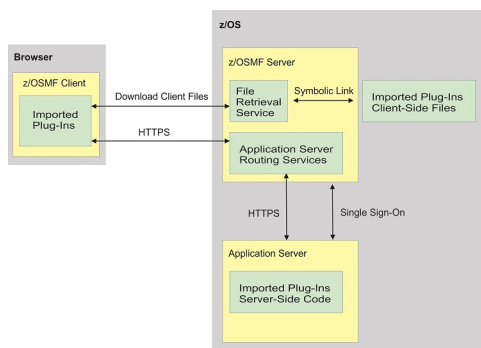


Figure 10. Process for routing requests and responses between application servers

As depicted in Figure 10 on page 33, the process of routing requests and responses for plug-ins where the client and server-side code are on different application servers is as follows:

1. Use the z/OSMF Import Manager task to import your plug-in into z/OSMF, and to associate each task in the plug-in with an application server. For instructions, see [“Adding your applications to z/OSMF” on page 1268](#).
2. During the import process, z/OSMF core creates symbolic links to the client-side code for your application and stores those links in the z/OSMF file system on the z/OSMF server.
3. When the import process completes, z/OSMF core adds the tasks included in your plug-in to the z/OSMF desktop.
4. When a user selects your task in the z/OSMF desktop, z/OSMF core submits an HTTPS request to the file retrieval service to retrieve the client-side files and the browser downloads those files.
5. When the user performs an action that requires the task to interact with the server-side code, the client-side code for the task submits an HTTPS request to the application server routing interface, and that interface routes the request to the application server that is associated with the task.
6. The server-side code for your task processes the request and submits an HTTPS response to the application server routing interface, and that interface routes the response to the client.

The z/OSMF server and the application server that hosts the server-side code for your plug-in (referred to as the *target application server*) can reside on the same system or on different systems. To enable single sign-on between the servers, ensure that the servers share the same Lightweight Third Party Authentication (LTPA) key files.

Operations provided through the application server routing services

[Table 8 on page 34](#) lists the operations that the application server routing services provide.

Table 8. Operations provided through the application server routing services.

Operation	HTTP method and URI path
“Retrieve data from an application server” on page 36	GET /zosmf/externalgateway/system?content=<http-content>
“Update data for an application server” on page 40	POST /zosmf/externalgateway/system PUT /zosmf/externalgateway/system
“Delete data from an application server” on page 43	DELETE /zosmf/externalgateway/system?content=<http-content>

Required authorizations

The user must be logged into z/OSMF. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for HTTP response data. If the client requests for the application server routing service to add additional information to the response from the target application server, the service wraps the response in the following JSON object. Otherwise, the service returns only the response from the target application server.

```
{
  "primaryAPIVersion": "primary-API-version",
  "systemsOutput": {
    "systemOutput": "system-output",
    "rc": "return-code",
    "error": { "msgid": "message-ID", "msgtxt": "message-text" },
    "systemName": "system-name"
  },
  "numOfSystems": "total-systems"
}
```

where

primary-API-version

Version of the application server routing services interface on the z/OSMF server.

systemsOutput

Contains a set of attributes that provide different information about the response from the target application server.

system-output

Contains the response from the target application server.

return-code

Contains the return code provided by the target application server. The return code can be one of the following values:

OK

Success.

HttpConnectionFailed

The HTTPS connection failed. Typically, this error occurs when the target application server is unavailable or a network error has occurred.

HttpConnectionTimedOut

The HTTPS request did not complete in the time allotted.

CertificateError

The certificate for the target application server is not trusted.

InvalidLogin

The login credentials for the target application server are not valid.

FailedWithMessage

The request was successful; however, an internal error occurred on the target application server.

UnexpectedFailure

An unexpected error occurred.

error

If an error occurred with the request, the error attribute contains the message ID (msgid) and message text (msgtxt) for the message that was issued. Otherwise, this attribute is *null*.

system-name

Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the target application server.

total-systems

Value is set to *1* because the HTTPS request can be sent to only one application server at a time.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request is not authorized to use the service or did not authenticate with z/OSMF, or single sign-on is not enabled between the z/OSMF server and the target application server.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the application server routing services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Retrieve data from an application server

You can use this operation to request that z/OSMF route a retrieve data request to the application server where the server-side code for your plug-in resides.

HTTP method and URI path

```
GET /zosmf/externalgateway/system?content=<http-content>
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 9 on page 36](#) lists the parameters that are supported for this request.

Important: If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *AppServer#1*, specify *AppServer%231*.

Table 9. Supported input parameters for the application server routing services		
Parameter	Required	Description
target	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() { var target = window.frameElement.getAttribute("target"); var remoteURL = "/zosmf/externalgateway/system? content={ 'target': '" + target + "', 'resourcePath': '/ testApp' }"; }</pre>
resourcePath	Yes	Path to the service that will process the request.
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.

Table 9. Supported input parameters for the application server routing services (continued)		
Parameter	Required	Description
wrapped	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see “Content type used for HTTP response data” on page 34 .
binary	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Application server routing services” on page 33](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 35](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 34](#).

Example 1: Retrieve wrapped data from an application server

To retrieve wrapped data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp"} HTTP/1.1
Host: appname.yourco.com
```

Figure 11. Sample request to retrieve wrapped data from an application server

A sample response is shown in [Figure 12 on page 38](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
    {
      "systemOutput":
        {
          "items":[
            {
              "object-ID":"objectA",
              "attribute1":"value1",
              "attribute2":"value2",
              "attribute3":"value3",
              "attribute4":"value4",
              "attribute5":"value5",
            },
            {
              "object-ID":"objectB",
              "attribute1":"value6",
              "attribute2":"value7",
              "attribute3":"value8",
              "attribute4":"value9",
              "attribute5":"value10",
            }
          ],
          "rc":"Ok",
          "error":null,
          "systemName":"appServer1"
        },
      "numOfSystems":1
    }
}
```

Figure 12. Sample response for retrieving wrapped data from an application server

Example 2: Retrieve unwrapped data from an application server

To retrieve unwrapped data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp",
"wrapped":"N"} HTTP/1.1
Host: appname.yourco.com
```

Figure 13. Sample request to retrieve unwrapped data from an application server

A sample response is shown in [Figure 14 on page 39](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
{
  "items": [
    {
      "object-ID": "objectA",
      "attribute1": "value1",
      "attribute2": "value2",
      "attribute3": "value3",
      "attribute4": "value4",
      "attribute5": "value5",
    },
    {
      "object-ID": "objectB",
      "attribute1": "value6",
      "attribute2": "value7",
      "attribute3": "value8",
      "attribute4": "value9",
      "attribute5": "value10",
    }
  ]
}
```

Figure 14. Sample response for retrieving unwrapped data from an application server

Example 3: Retrieve binary data from an application server

To retrieve binary data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp",
"binary":"Y"} HTTP/1.1

Host: appname.yourco.com
```

Figure 15. Sample request to retrieve binary data from an application server

A sample response is shown in [Figure 16 on page 40](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
01111011 00001101 00001010 00100000 00100000 00100010 01101001 01101000 01100101 01101101
01110011 00100010 00111010 01011011 00001101 00001010 00100000 00100000 01111011 00001101
00001010 00100000 00100000 00100000 00100000 00100010 01101111 01100010 01101010 01100101
01100011 01101010 00101101 01001001 01000100 00100010 00100010 00111010 00100010 01101111 01100010
01101010 01100101 01100011 01101000 01000001 00100010 00001101 00001010 00100000 00100000
00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101
01110100 01100101 00110001 00100010 00111010 00100010 01110110 01100001 01101100 01110101
01100101 00110001 00100010 00101100 00001101 00001010 00100000 00100000 00100000 00100000
00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101
00110010 00100010 00111010 00100010 01110110 01100001 01101100 01110101 01100101 00110010
00100010 00101100 00001101 00001010 00100000 00100000 00100000 00100000 00100010 01100001
01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101 00110011 00100010
00111010 00100010 01110110 01100001 01101100 01110101 01100101 00110011 00100010 00101100
00001101 00001010 00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100
01110010 01101001 01100010 01110101 01110100 01100101 00110100 00100010 00111010 00100010
01110110 01100001 01101100 01110101 01100101 00110100 00100010 00101100 00001101 00001010
00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001
01100010 01110101 01110100 01100101 00110100 00110100 00100010 00111010 00100010 01110110
01100101 01101001 01100010 01110101 01100101 00110100 00100010 00110011 00100010 00111010
01110110 01100001 01101100 01110101 01100101 00111000 00100010 00101100 00001101 00001010
00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001
01100010 01110101 01110100 01100101 00110100 00100010 00111010 00100010 01110110 01100001
01101100 01110101 01100101 00111001 00100010 00101100 00001101 00001010 00100000 00100000
00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101
01110100 01100101 00110101 00100010 00111010 00100010 01110110 01100001 01101100 01110101
01100101 00110001 00110001 00110000 00101100 00001101 00001010 00100000 00100000 01111101
01011101 00001101 00001010 01111101
```

Figure 16. Sample response for retrieving binary data from an application server

Update data for an application server

You can use this operation to request that z/OSMF route an update data request to the application server where the server-side code for your plug-in resides.

HTTP method and URI path

```
POST /zosmf/externalgateway/system
PUT /zosmf/externalgateway/system
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include a JSON object that describes the objects to be created or modified on the target application server. [Table 10 on page 41](#) lists the supported parameters.

Table 10. Supported input parameters for the application server routing services		
Parameter	Required	Description
target	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() { var target = window.frameElement.getAttribute("target"); var remoteURL = "/zosmf/externalgateway/system? content= {'target':'" + target + "', 'resourcePath': '/ testApp'}"; }</pre>
resourcePath	Yes	Path to the service that will process the request.
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
wrapped	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see “Content type used for HTTP response data” on page 34.
binary	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.

Table 10. Supported input parameters for the application server routing services (continued)

Parameter	Required	Description
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Required authorizations

See [“Application server routing services” on page 33.](#)

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 35.](#)

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 34.](#)

Example

To add *objectC* on the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
POST /zosmf/externalgateway/system HTTP/1.1
Host: appname.yourco.com
{"target":"appServer1","resourcePath":"/testApp/objectC","content":{"attribute1":"value11","attribute2":"value12","attribute3":"value13","attribute4":"value14","attribute5":"value15"}}
```

Figure 17. Sample request to update data on an application server

A sample response is shown in [Figure 18 on page 42.](#)

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "result":"success"
    },
    "rc":"Ok",
    "error":null,
    "systemName":"appServer1"
  },
  "numOfSystems":1
}
```

Figure 18. Sample response for updating data on an application server

Delete data from an application server

You can use this operation to request that z/OSMF route a delete data request to the application server where the server-side code for your plug-in resides.

HTTP method and URI path

```
DELETE /zosmf/externalgateway/system?content=<http-content>
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 11 on page 43](#) lists the parameters that are supported for this request.

Important: If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *AppServer#1*, specify *AppServer%231*.

Table 11. Supported input parameters for the application server routing services

Parameter	Required	Description
target	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() { var target = window.frameElement.getAttribute("target"); var remoteURL = "/zosmf/externalgateway/system? content= {'target':'" + target + "', 'resourcePath': '/ testApp'}"; }</pre>
resourcePath	Yes	Path to the service that will process the request.
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.

Table 11. Supported input parameters for the application server routing services (continued)

Parameter	Required	Description
wrapped	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see “Content type used for HTTP response data” on page 34.
binary	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Application server routing services”](#) on page 33.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 35.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 34.

Example

To remove *objectA* from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
DELETE /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp/objectA",
"timeout":"30000"} HTTP/1.1
```

Host: appname.yourco.com

Figure 19. Sample request to delete data from an application server

A sample response is shown in [Figure 20 on page 45](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "result":"success"
    },
    "rc":"Ok",
    "error":null,
    "systemName":"appServer1"
  },
  "numOfSystems":1
}
```

Figure 20. Sample response for deleting data from an application server

Cloud provisioning services

The cloud provisioning services are a set of application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to perform software provisioning for IBM Cloud Provisioning and Management for z/OS. This includes creating instances of z/OS and IBM middleware, such as IBM Customer Information Control System (CICS), IBM Db2, IBM Information Management System (IMS), IBM MQ, and IBM WebSphere Application Server (WAS), and creating middleware resources, such as MQ queues, CICS regions, and Db2 databases. This capability makes it possible for consumers to quickly provision and deprovision an environment as needed.

Getting started

The security administrator defines the various roles that are required, such as the domain administrator, network administrator, approvers, and consumers. For more information about setting up security for cloud provisioning, see [Cloud provisioning services](#) in *IBM z/OS Management Facility Configuration Guide*.

Using the Cloud Provisioning tasks, your system programmers and application programmers can perform the following actions:

- System programmers:
 - Define the cloud domain (systems), administrators for the domain, and classes of users (tenants) for the domain.
 - Prepare software services templates, which are used to provision z/OS software. Service providers add templates, associate tenants with the templates, create resource pools for the templates, test the templates, then publish them to make them available for consumers.
- System programmers or application programmers:
 - Provision software from templates, creating software services instances.
 - Manage software services instances.

For an illustration of cloud provisioning, see [Figure 21 on page 46](#).

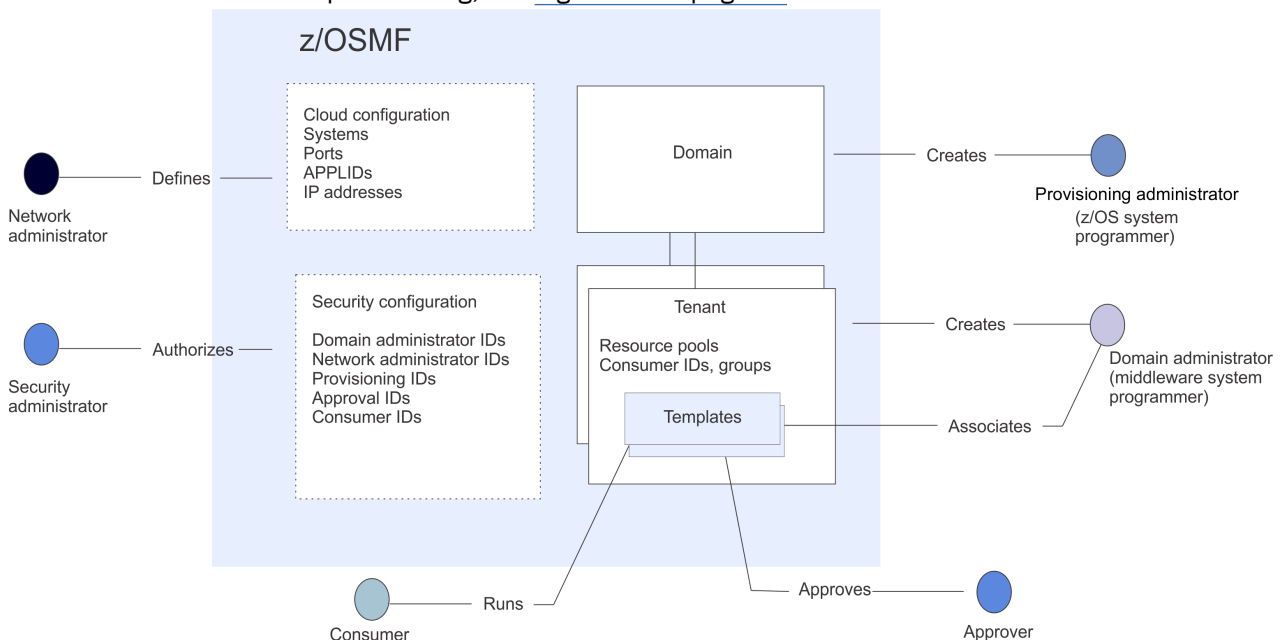


Figure 21. Cloud Provisioning Summary

Domains define the scope of cloud provisioning

A cloud provisioning domain defines a system or set of systems in the sysplex. The systems in the domain must be included in the group of systems named IYUCLOUD in the **Systems** task of the **z/OSMF Settings** category.

A provisioning administrator, typically a z/OS system programmer, decides which system or systems (LPARs) are used for provisioning and creates a domain. If the domain extends beyond one sysplex, the provisioning administrator configures a primary z/OSMF system for communicating with secondary z/OSMF systems.

To help you get started quickly, a default domain is provided. The default domain is fully operational without any further configuration, and is accessible to any z/OSMF administrator. A default tenant is associated with the default domain.

When a domain includes more than one system, the domain administrator can specify:

- The systems that are to be used as potential targets for provisioning
- How the target system should be selected when the software service is provisioned: either automatically, by z/OSMF, or manually, by the consumer
- That the instance can be relocated to a system in the domain other than the system it was originally provisioned on. The instance can run on only one system at a time.

Templates guide provisioning

To make an environment available to consumers as a software service, a domain administrator creates and configures a software services template. The template describes what is provisioned. For example, a template might request that a Db2 subsystem be deployed onto a z/OS system with three databases, or might create a set of CICS regions.

To provision the middleware, templates start and run z/OSMF workflows. A template includes a workflow definition file, along with other files, including a file that defines input variables for the workflow, and a file that defines actions that can be used against the provisioned software.

The template might need to be customized for the installation – for example, to conform with naming standards in your company. You might modify variables that are input to the workflow, or use a properties file that is provided with the template to configure the provisioned software. For information about customization, you typically refer to documentation that is included with the template by the software provider. In addition, the domain administrator:

- Adds the software services template to a tenant.
- Connects the template to network, storage, WLM, and LPAR resource pools. Resource pools are sets of z/OS resources that are required by the z/OS software service, for example, ports, IP addresses, or APPLIDs.

When a template requires resource pools, for example, when you want to dynamically allocate ports to provisioned software instances, the network and WLM resource pool administrators (typically z/OS system programmers) use the appropriate z/OSMF tasks to complete the resource pools.

Offering self-service provisioning to a development team might require that some steps in the template, or certain actions, run under automation IDs. Any use of these user IDs in a template must be approved. Approval records are created for a template when a workflow or action definition file contains an element that identifies a user ID under which a workflow step or action is to be performed. (The workflow element is `runAsUser` ID, and the ID is sometimes referred to as a `runAsUser` ID). Approval records can also be defined for the template in general, and for a domain. Approval records must be approved by the approvers (typically identified by user ID) before the template can be tested or published.

The domain administrator tests the template to ensure that it successfully provisions the software, that is, creates the environment. Software that is provisioned from a template is known as a software services instance. (Note that this is different than a software instance that you manage with the Software Management task. A *software instance* is a collection of data sets containing installed software, and other

data sets that may be associated with that installed software.) You manage a software services instance by using actions such as **Remove** and **deprovision**.

Publishing the template makes it available to consumers in the tenant – the application developers who require the new environment.

Summary

The terms that you need to understand for provisioning and managing provisioned software are defined here.

Resources

The following are the key resources in the Cloud Provisioning tasks.

Table 12. Resources for Cloud Provisioning	
Resource	Description
<i>Domain</i>	<p>Defines the management scope for tenants, services, and resource pools.</p> <p>A domain consists of one or more z/OS systems. A domain can include z/OS systems from more than one sysplex.</p> <p>A z/OS system can be in a single domain or in multiple domains that are managed by a single instance of z/OSMF. A cloud domain is defined by a z/OS system programmer who acts as the <i>provisioning administrator</i>. Each cloud domain is assigned one or more middleware system programmers who act as domain administrators.</p> <p>A base z/OSMF configuration includes one domain by default – the default domain.</p>
<i>Resource pool</i>	<p>Identifies the z/OS resources that are required by a z/OS software service. In a cloud domain with multiple tenants, the resource pool defines the scope of resource sharing and resource isolation. For example, a resource pool can define a range of dedicated IP addresses or ports for each tenant.</p> <p>A base z/OSMF configuration includes one resource pool by default – the default domain shared resource pool.</p>
<i>Tenant</i>	<p>Defines the group of users who have the authority to provision software instances.</p> <p>A tenant consists of a user or group of users that have contracted for the use of specified services and pooled z/OS resources that are associated with the services in a domain.</p> <p>A base z/OSMF configuration includes one tenant by default – the default tenant.</p>

User roles

The following are the key roles in the Cloud Provisioning tasks.

Table 13. User roles for Cloud Provisioning		
Role	Performer	Description
<i>Provisioning administrator</i>	z/OS system programmer	Defines the cloud domains and the associated system resources for the cloud. The provisioning administrator also designates one or more users as domain administrators.
<i>Domain administrator</i>	Middleware system programmer	Manages a domain. The domain administrator is responsible for defining services, tenants, and resource pools for the domain, and managing the relationship across tenants, services, and resource pools.

Table 13. User roles for Cloud Provisioning (continued)		
Role	Performer	Description
<i>Resource pool networking administrator</i>	Network administrator	Manages the resource pool for the networking resources in the cloud, such as network configuration policies.
<i>Resource pool WLM administrator</i>	Performance administrator	Manages the resource pool for the WLM resources in the cloud, such as WLM policies.
<i>Security administrator</i>	Security administrator	Maintains the installation's security manager, such as RACF.
<i>Template approver</i>	System programmer or security administrator	Responsible for approving the pending approval records that are associated with the template.
<i>Consumer</i>	Application programmer	Has access to the software services and resource pools for a tenant. This user can provision a software services instance by using a software services template, and can manage the lifecycle of a software services instance.

Objects

The following are some basic objects that you work with in the Cloud Provisioning tasks.

Table 14. Objects for Cloud Provisioning	
Object	Description
<i>Instance, or software services instance</i>	Represents software that is provisioned by using templates.
<i>Template, or software services template</i>	Represents z/OS or z/OS middleware, or a z/OS middleware resource service. A template consists of workflows and input variables that can be used to provision z/OS software, actions that can be used with the provisioned software (the instance), and documentation.

Authorization requirements

Use of the cloud provisioning services APIs requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

In addition, the user's z/OS user ID may need access to other resources, including those that define roles. The specific requirements for each cloud provisioning service are described in the topic for that service. For a summary of resources related to roles, see [Table 15 on page 49](#).

Table 15. SAF resources for Cloud Provisioning Roles			
Role	Class	Resources	Access
Provisioning administrator	ZMFCLLOUD	<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT.saf_cloud_groupID_prefix	READ
Domain administer	ZMFCLLOUD	<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT.domainGroupID	READ
Domain approver	ZMFCLLOUD	<SAF-prefix>.ZOSMF.TEMPLATE.APPROVERS.domainGroupID	READ

Table 15. SAF resources for Cloud Provisioning Roles (continued)			
Role	Class	Resources	Access
Template runAsUser ID	ZMFCLOUD	<SAF-prefix>.ZOSMF.TEMPLATE. RUNASUSERS.domainGroupID.templateID	READ ¹
Template approver	ZMFCLOUD	<SAF-prefix>.ZOSMF.TEMPLATE. APPROVERS.domainGroupID.templateID	READ
Tenant	ZMFCLOUD	<SAF-prefix>.ZOSMF.PROVISIONING. RESOURCE_MANAGEMENT.tenantGroupID	READ
Resource pool network administrator	ZMFCLOUD	<SAF-prefix>.ZOSMF. RESOURCE_POOL.NETWORK.domainGroupID	READ
Resource pool WLM administrator	ZMFCLOUD	<SAF-prefix>.ZOSMF. RESOURCE_POOL.WLM.domainGroupID	READ

¹ Successful READ attempts for the <SAF-prefix>.ZOSMF.TEMPLATE.
RUNASUSERS.domainGroupID.templateID resource are audited. Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow fails.

For details about security for the cloud provisioning roles, see [Cloud provisioning services in IBM z/OS Management Facility Configuration Guide](#).

For information about how to prepare software for provisioning through the REST APIs or the z/OSMF Cloud Provisioning tasks, including the format of the file for defining actions, see [Preparing software to exploit cloud provisioning in IBM z/OS Management Facility Programming Guide](#).

Using the Swagger interface

You can use the Swagger interface to display information about the IBM Cloud Provisioning and Management for z/OS REST APIs.

For more information, see [Using the z/OSMF REST services in IBM z/OS Management Facility Programming Guide](#).

Resource pool services

The resource pool services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to obtain and release network or WLM resources from the network or WLM resource pool that was defined in support of IBM Cloud Provisioning and Management for z/OS. These REST services are intended to be invoked from a workflow step during cloud provisioning and are not available for general use outside of the scope of cloud provisioning.

Table 16 on page 51 lists the operations that the resource pool services provide.

See also:

- [“Get a resource pool” on page 158](#)
- [“Get a domain resource pool” on page 164](#)
- [“List the resource pools” on page 173](#)
- [“List domain resource pools” on page 179](#)

Resource pool services

Table 16. z/OSMF resource pool services: operations summary	
Operation name	HTTP method and URI path
“Obtain an IP address” on page 52	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/obtain
“Release an IP address” on page 58	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/release
“Obtain a port” on page 61	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/obtain
“Release a port” on page 65	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/release
“Obtain a SNA application name” on page 67	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/obtain
“Release a SNA application name” on page 70	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/release
“Add a classification rule” on page 72	POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/add
“Remove a classification rule” on page 74	POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/remove
“Get data set attributes” on page 76	GET /zosmf/resource-mgmt/rest/<version>/rdp/storage/dataset-attr/<tenant-id>/<template-name>/<registry-uuid>
“Create an LPAR resource pool entry” on page 78	PUT /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/
“Modify an LPAR resource pool entry” on page 82	POST /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/<lpar-pool-id>
“Delete an LPAR resource pool entry” on page 86	DELETE /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/<lpar-pool-id>
“Retrieve a list of LPAR resource pool entries” on page 87	GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/
“Retrieve the properties of an LPAR resource pool entry” on page 91	GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/<lpar-pool-id>

Table 16. z/OSMF resource pool services: operations summary (continued)

Operation name	HTTP method and URI path
“Obtain an LPAR resource pool entry” on page 95	POST /zosmf/resource-mgmt/rest/<version>/rdp/lpar/actions/obtain
“Release an LPAR resource pool entry” on page 100	POST /zosmf/resource-mgmt/rest/<version>/rdp/lpar/actions/release

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, a network administrator for APIs related to network resources, or a WLM administrator for APIs related to WLM resources, in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 204 No Content

The request succeeded.

HTTP 500 Server error

The server encountered an error when it processed the request.

Obtain an IP address

Use this operation to obtain an IP address from a resource pool that has a configured network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/obtain
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation obtains an IP address from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in an IP address being obtained.

Request content

The request content is expected to contain a JSON object that describes the IP address to be obtained. See [Table 17 on page 53](#).

Table 17. Request content for the obtain IP address request

Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none">• Provisioning workflows: <code>\${_workflow-registryID}</code>• Action workflows: <code>\${_workflow-parentRegistryID}</code> Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, <code>\${_workflow-templateID}</code> .
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-templateName}</code> .
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-tenantID}</code> .
network-parms	JSON object	Required	Network. parameters for the request. See Table 18 on page 53 .

Table 18. Network parameters fields

Field	Type	Required or optional	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable <code>\${_workflow-softwareServiceInstanceName}</code> for this parameter.
usage-type	String	Optional	Used as a filter. If not specified, only network resource pools without a usage type match. If specified, must match the usage type in the network resource pool definition in the Network Configuration Assistant task.

Table 18. Network parameters fields (continued)

Field	Type	Required or optional	Description
ipaddr	String	Required	<p>IP address. The value can be:</p> <p>A specific IP address Provision this IP address. The IP address must fit within the available IP address allocation range. Available ranges are associated with the targeted network resource pool, and match the provided usage type.</p> <p>any4 Provision any available Ipv4 address, from the available range.</p> <p>any6 Provision any available IPv6 address, from the available range.</p>
system-name	String	See description	Specifies the target system that the resource will be provisioned on. Required if there is more than one system in the network resource pool. Derived from a workflow internal variable, <code>\${_workflow-systemName}</code> .
deployment-id	String	Optional	Workflow-defined string token, used to group all provisioned resources with a server instance.
host-name	String	Optional	Indicates that the domain name server that is configured to the associated IP address allocation range is to be updated with the IP address and the concatenation of the host name and the zone name from the IP address allocation range. Requires a domain name server object to be configured by the network administrator.
system-list[]	String	Optional	A list of target systems for provisioning. If you omit this field and the system-name field, all systems in the network resource pool are targets. See Table 20 on page 56 for details on the values.

Table 18. Network parameters fields (continued)

Field	Type	Required or optional	Description										
recovery-method	String	Optional	<p>Defines the availability characteristics of an IP address under high availability recovery situations. The following values are valid:</p> <p>MANUAL_DISRUPTIVE Does not allow the IP address to be moved non-disruptively to another TCP/IP stack. Connections to this VIPA are broken if the VIPA is moved. Only one application can simultaneously use this IP address.</p> <p>MANUAL_NONDISRUPTIVE Allows the IP address to be moved non-disruptively to another TCP/IP stack. Only one application can simultaneously use this IP address, unless the application-owned field has a value of false.</p> <p>DYNAMIC The IP address is distributed and therefore available to multiple equivalent applications simultaneously. There is no need to move the IP address for recovery situations.</p> <p>The default value is dependent upon the deployment type of resource pool, represented by the Workload Deployment Type attribute of the Network Resource Pool in Network Configuration Assistant.</p> <table><tr><td colspan="2">Table 19. Default values for recovery-method</td></tr><tr><th>Workload Deployment Type</th><th>recovery-method Default</th></tr><tr><td>SINGLE_SYSTEM</td><td>MANUAL_DISRUPTIVE</td></tr><tr><td>MOVABLE</td><td>MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.</td></tr><tr><td>CLUSTER</td><td>DYNAMIC</td></tr></table>	Table 19. Default values for recovery-method		Workload Deployment Type	recovery-method Default	SINGLE_SYSTEM	MANUAL_DISRUPTIVE	MOVABLE	MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.	CLUSTER	DYNAMIC
Table 19. Default values for recovery-method													
Workload Deployment Type	recovery-method Default												
SINGLE_SYSTEM	MANUAL_DISRUPTIVE												
MOVABLE	MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.												
CLUSTER	DYNAMIC												

Table 18. Network parameters fields (continued)

Field	Type	Required or optional	Description
application-owned	String	Optional	<p>Defines the ownership characteristics of an IP address with regard to applications. The following values are valid:</p> <p>"true"</p> <p>The IP address is owned by the single application that is currently bound to the IP address. Only one application can simultaneously use this IP address. This is the default value. Requests containing the same deployment-id from the same template instance (same job-name) will provision different IP addresses each time. Requests from a different template instance (different job-name) containing the same deployment-id as previous requests will return the same IP address returned to those previous requests and increment a use count. When deprovisioning, the use count must reach zero before the IP address is returned to the pool.</p> <p>"false"</p> <p>The IP address is not owned by any single application. Multiple applications can use the IP address simultaneously. A value of false is only valid when the recovery-method value is MANUAL_NONDISRUPTIVE.</p> <p>Note: This property is a string value (not Boolean).</p>
job-name	String	Required	Job name associated with the provisioned instance.
requires-zcx-addr	boolean	Optional	Indicates whether the resource pool needs to support z/OS Container Extensions (zCX).

Table 20. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 57](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 21. HTTP error response codes for an obtain IP address request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 22 on page 57](#).

Table 22. Response from an obtain IP address request		
Field	Type	Description
id	String	Identifier of the IP address.
ipaddr	String	IP address returned from the Network Configuration Assistant task.

Example HTTP interaction

In [Figure 22 on page 58](#), a request is submitted to obtain an IP address.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/ip/obtain
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-params": {
    {
      "name": "CICSA IP",
      "usage-type": "internal",
      "ipaddr": "any4",
      "system-list": [
        {
          "sysplex-name": "LOCAL",
          "sysplex-node-name": "MVS",
          "system-nickname": "MVS"
        }
      ]
    },
    "deployment-id": "CICSBasic",
    "host-name": "myHostName",
    "recovery-method": "DYNAMIC",
    "application-owned": true,
    "requires-zcx-addr": true
  }
}
```

Figure 22. Sample request to obtain an IP address, with the request body

The following is the response body for the example obtain IP address request.

```
{
  "id": "101",
  "ipaddr": "192.168.1.1"
}
```

Release an IP address

Use this operation to release an IP address from a network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/release
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation releases an IP address from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in an IP address being released.

Request content

The request content is expected to contain a JSON object that describes the IP address to be released. See [Table 23 on page 59](#).

Table 23. Request content for the release IP address request

Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> Provisioning workflows: \${_workflow-registryID} Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-tenantID}.
network-parms	JSON object	Required	Network parameters for the request. See Table 24 on page 59 .

Table 24. Network parameters fields

Field	Type	Required or optional	Description
ip-id	String	Required	Identifier of the IP address. This is returned as the id property in an Obtain an IP address request.
system-list[]	String	Optional	A list of target systems for deprovisioning, When specified, system-list must contain only one system or all systems in the network resource pool. When system-list is omitted and system-name is not present, the resource will be deprovisioned on each system in the network resource pool. See Table 25 on page 59 for details on the values.

Table 25. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 26. HTTP error response codes for a release IP address request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Example HTTP interaction

In [Figure 23 on page 60](#), a request is submitted to release an IP address.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/ip/release
```

```
{
  "registry-uuid": "E1E2A1C4",
  "tenant-id": "IYU0AA",
  "network-params": {
    "ip-id": "1001",
    "system-list": [
      {
        "sysplex-name": "LOCAL",
        "sysplex-node-name": "MVS",
        "system-nickname": "MVS"
      }
    ]
  }
}
```

Figure 23. Sample request to release an IP address, with the request body

Obtain a port

Use this operation to obtain a port from a resource pool that has a configured network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/obtain
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation obtains a port from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a port being obtained.

Request content

The request content is expected to contain a JSON object that describes the port to be obtained. See [Table 27 on page 61](#).

Table 27. Request content for the obtain port request			
Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none">Provisioning workflows: \${_workflow-registryID}Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
network-parms	JSON object	Required	Network parameters for the request. See Table 28 on page 62 .

Table 28. Network parameters fields

Field	Type	Required or optional	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable \${_workflow-softwareServiceInstanceName} for this parameter.
usage-type	String	Optional	Used as a filter. If not specified, only network resource pools without a usage type match. If specified, must match the usage type in the network resource pool definition in the Network Configuration Assistant task.
port	String	Optional	Request port number. If port is not specified, a port is provisioned from available port allocation ranges of the specified transport. If port is specified, it must fit within an available range. Available ranges are those which are associated with the targeted network resource pool and match the provided usage type.
job-name	String	Required	Job name associated with the provisioned instance.
system-name	String	Optional	System name. Derived from a workflow internal variable, \${_workflow-systemName}.
deployment-id	String	Optional	Workflow-defined string token, used to group all provisioned resources with a server instance.
host-name	String	Optional	Host-name for the Network Configuration Assistant task.
system-list[]	String	Optional	A list of target systems for provisioning. If you omit this field and the system-name field, all systems in the network resource pool are targets. See Table 29 on page 63 .
is-port-shared	String	Optional	Indicates whether the port should be capable of being shared with other servers on the same system that are listening on the same IP address and port. Used to enable same-system coexistence or when multiple equivalent servers normally run simultaneously on the same system bound to the same IP address and port.
is-port-distributed	String	Optional	Indicates whether the port is used for workload distribution, as follows: true The port will be used in conjunction with provisioned distributed or group DVIPAs (IP addresses provisioned with a value of DYNAMIC for the recovery-method field) when creating the TCP/IP VIPADISTRIBUTE statement for workload distribution among the clustered instances. These are ports that the server application will use when binding its listening sockets to the distributed or group DVIPA. false The port is not used for workload distribution.

Table 28. Network parameters fields (continued)

Field	Type	Required or optional	Description
intent	String	Optional	<p>Indicates the intent, as follows:</p> <p>shared The port number can be reused for multiple port provisioning requests on the same TCP/IP stack. This is the default value.</p> <p>exclusive The port provisioning request is given a port number that has not been previously provisioned on this TCP/IP stack. Moreover, subsequent port provisioning requests do not return the same port number on the same TCP/IP stack, as long as this port remains provisioned.</p> <p>workloadExclusive The port provisioning request is given port numbers that are unique for each port provisioning request within the same cluster, regardless of the system or stack it is provisioned on.</p>

Table 29. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 64](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 30. HTTP error response codes for an obtain port request

HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.

Table 30. HTTP error response codes for an obtain port request (continued)	
HTTP error status code	Description
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 31 on page 64](#).

Table 31. Response from an obtain port request		
Field	Type	Description
id	String	Identifier of the port.
port	String	Port number.

Example HTTP interaction

In [Figure 24 on page 64](#), a request is submitted to obtain a port.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/port/actions/obtain
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-parms": {
    {
      "name": "PortForCics1",
      "port": "80",
      "usage-type": "Internal",
      "job-name": "WLP001",
      "deployment-id": "CICSBasic",
      "host-name": "myHostName",
      "is-port-shared": true,
      "is-port-distributed": true,
      "intent": "workloadExclusive",
      "system-list": [
        {
          "sysplex-name": "LOCAL",
          "sysplex-node-name": "MVS",
          "system-nickname": "MVS"
        }
      ]
    }
  ]
}
```

Figure 24. Sample request to obtain a port, with request body

The following is the response body for the example obtain port request.

```
{
  "id": "82346",
  "port": "80",
}
```

Release a port

Use this operation to release a port from a network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/release
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation releases a port from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a port being released.

Request content

The request content is expected to contain a JSON object that describes the port to be released. See [Table 32 on page 65](#).

Table 32. Request content for the release port request			
Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none">Provisioning workflows: \${_workflow-registryID}Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
network-parms	JSON object	Required	Network parameters for the request. See Table 33 on page 66 .

Table 33. Network parameters fields			
Field	Type	Required or optional	Description
port-id	String	Required	Identifier of the port. This is returned as the id property in an Obtain a port request.
system-list[]	String	Optional	A list of target systems for deprovisioning. When specified, system-list must contain only one system or all systems in the network resource pool. When system-list is omitted and system-name is not present, the resource will be deprovisioned on each system in the network resource pool. See Table 34 on page 66 for details on the values.
system-name	String	Optional	Specifies the target system that the resource will be deprovisioned from.

Table 34. Systems field			
Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 35. HTTP error response codes for a release port request	
HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Example HTTP interaction

In [Figure 25 on page 67](#), a request is submitted to release a port.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/port/actions/release
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IYU0AA",
  "network-params" :
  {
    "port-id" : "1001",
    "system-name": "SY1"
  }
}
```

Figure 25. Sample request to release a port, with the request body

Obtain a SNA application name

Use this operation to obtain a SNA application name from a resource pool that has a configured network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/obtain
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation obtains a SNA application name from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a SNA application name being obtained.

Request content

The request content is expected to contain a JSON object that describes the SNA application name to be obtained. See [Table 36 on page 68](#).

Table 36. Request content for the obtain SNA application name request

Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> Provisioning workflows: <code>\${_workflow-registryID}</code> Action workflows: <code>\${_workflow-parentRegistryID}</code> Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, <code>\${_workflow-templateID}</code> .
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, <code>\${_workflow-templateName}</code> .
tenant-id	String	Required	Name of the tenant that is associated with the resource pool. Derived from a workflow internal variable, <code>\${_workflow-tenantID}</code> .
network-parms	JSON object	Required	Network parameters for the request. See Table 37 on page 68 .

Table 37. Network parameters fields

Field	Type	Optional/ Required	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable <code>workflow internal variable, \${_workflow-softwareServiceInstanceName}</code> for this parameter.
deployment-id	String	Optional	Workflow-defined string token.
sna-application-name	String	Required	A name for the SNA application. Derived from the workflow internal variable <code>\${_workflow-softwareServiceInstanceName}</code> .

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 69](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 38. HTTP error response codes for an obtain SNA application name request</i>	
HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 39 on page 69](#).

<i>Table 39. Response from an obtain SNA application name request</i>		
Field	Type	Description
id	String	Identifier of the SNA application name. Needed for the release call, as the value for the appl-name-id property.
appl-name	String	Required. Application name from the network resource pool in the Network Configuration Assistant task.

Example HTTP interaction

In [Figure 26 on page 69](#), a request is submitted to obtain a SNA application name.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/snaapplname/actions/obtain

{
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-parms" :
  {
    "name": "CICSA APPLID",
    "deployment-id": "CICSBasic",
    "sna-appl-name": "CICSA001"
  }
}
```

Figure 26. Sample request to obtain a SNA application name, with request body

The following is the response body for the example obtain SNA application name request.

```
{
  "id": 82346,
  "appl-name": "CICSC10"
}
```

Release a SNA application name

Use this operation to release a SNA application name from a network resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/release
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation releases a SNA application name from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a SNA application name being released.

Request content

The request content is expected to contain a JSON object that describes the SNA application name to be released. See [Table 40 on page 70](#).

Table 40. Request content for the release SNA application name request			
Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none">Provisioning workflows: \${_workflow-registryID}Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
network-parms	JSON object	Required	Network parameters for the request. See Table 41 on page 71 .

Table 41. Network parameters fields

Field	Type	Optional/ Required	Description
appl-name-id	String	Required	Identifier of the SNA application name. This is returned as the id property in an Obtain a SNA application name request.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 42. HTTP error response codes for a release SNA application name request

HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Example HTTP interaction

In [Figure 27 on page 71](#), a request is submitted to release a SNA application name.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/snaapplname/release
{
  "tenant-id": "IYU0AA",
  "network-parms":
  {
    "appl-name-id": "82346"
  }
}
```

Figure 27. Sample request to release a SNA application name

Add a classification rule

Use this operation to add a classification rule in a WLM Policy with service level agreement specified in a resource pool.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/add
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation adds a classification rule in a WLM policy for a middleware instance with a service level agreement specified in the tenant's resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a classification rule being added.

Request content

The request content is expected to contain a JSON object. See [Table 43 on page 72](#).

Table 43. Request content for the add classification rule request			
Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none">Provisioning workflows: \${_workflow-registryID}Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-name	String	Required	Name of the template that is associated with the tenant. Derived from a workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
wlm-parms	JSON object	Required	WLM parameters for the request. See Table 44 on page 73 .

Table 44. WLM parameters fields			
Field	Type	Optional/ Required	Description
qualifier	String	Required	The started task name. In most cases it can be derived from workflow internal variable \${_workflow-softwareServiceInstanceName}

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a WLM administrator in the domain that the tenant is associated with.

The user must also be the owner of the software services registry entry for the software services instance.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 45. HTTP error response codes for a release SNA application name request	
HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 46 on page 73](#).

Table 46. Response from an add classification rule request		
Field	Type	Description
cl-rule-id	String	Required. Identifier of the classification rule.
report-class-name	String	Report class that is associated with the resource pool.
service-class-name	String	Service class that is associated with the SLA that is defined in the resource pool.

Example HTTP interaction

In [Figure 28 on page 74](#), a request is submitted to add a classification rule.

```

POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/wlm/clrule/actions/add
{
  "registry-uuid": "E1E2A1C4",
  "template-name": "CICSBasic",
  "tenant-id": "IYU102",
  "resistry-id": "0d375584-305d-4bd5-b26e-88ac74c8171a",
  "wlm-parms":
  {
    "qualifier": "CICSA001"
  }
}

```

Figure 28. Sample request to add a classification rule

The following is the response body for the request:

```

{
  "cl-rule-id" : "82346",
  "report-class-name": "RPTCLASS",
  "service-class-name": "SCGOLD",
}

```

Remove a classification rule

Use this operation to remove a classification rule from a WLM Policy.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/remove
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation removes a classification rule from a WLM policy.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a classification rule being removed.

Request content

The request content is expected to contain a JSON object. See [Table 47 on page 75](#).

Table 47. Request content for the remove classification rule request

Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> Provisioning workflows: \${_workflow-registryID} Action workflows: \${_workflow-parentRegistryID} Required when provisioning network resources as part of a composite cluster template.
template-name	String	Required	Name of the template that is associated with the tenant. Derived from a workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
wlm-parms	JSON object	Required	WLM parameters for the request. See Table 48 on page 75 .

Table 48. WLM parameters fields

Field	Type	Optional/ Required	Description
cl-rule-id	String	Required	Returned by an Add Classification Rule request as the id property.

Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a WLM administrator in the domain that the tenant is associated with.

The user must also be the owner of the software services registry entry for the software services instance.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (No Content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 49. HTTP error response codes for a release SNA application name request	
HTTP error status code	Description
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

Table 49. HTTP error response codes for a release SNA application name request (continued)	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Example HTTP interaction

In Figure 29 on page 76, a request is submitted to add a classification rule.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/wlm/classification-rule/actions/remove
{
  "registry-uuid": "E1E2A1C4",
  "template-name": "CICSBasic",
  "tenant-id": "IYU102",
  "registry-id": "0d375584-305d-4bd5-b26e-88ac74c8171a",
  "wlm-parms":
  {
    "cl-rule-id" : 82346,
  }
}
```

Figure 29. Sample request to remove a classification rule

Get data set attributes

Use this operation to retrieve data set attributes.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/rdp/storage/dataset-attr/<tenant-id>/<template-name>/<registry-uuid>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<tenant-id>

Identifies the tenant that the data set attributes are associated with.

<template-name>

Identifies the template that the data set attributes are associated with.

<registry-uuid>

The UUID of the registry instance. This value is required only when using a composite cluster template.

Query parameters

dsn-type (optional)

The type of data set that will be allocated on the storage resource.

size (optional)

The size of the storage resource requested.

- If both *dsn-type* and *size* are omitted, all of the data set attribute information is returned for the storage pool.
- If only *dsn-type* is specified, all of the entries that have a matching *dsn-type* are returned for all *sizes* found.
- If only *size* is specified, all of the entries that have a matching *size* are returned for all *dsn-type* found.

Description

This operation retrieves data set attributes.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 77](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 50. HTTP error response codes for a get data set attributes request</i>	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains an array of data set attributes.

```

{
  "dataset-attributes-list" : [
    {
      "id": "RDPIDSPDS",
      "size": "SIZE",
      "type": "TYPE",
      "data-class": "DATACLAS",
      "storage-class": "STORCLAS",
      "management-class": "DATACLAS",
      "volser": "VOLSER",
      "description": "DESCRIPTION"
    }
  ]
}
Success: 200 (OK)

Error: 404 Not Found

```

Figure 30. Response from a get data set attributes request.

Example HTTP interaction

In Figure 31 on page 78, a request is submitted to retrieve a resource pool.

```

<rest>
<httpMethod>GET</httpMethod>
<uriPath substitution="true">/zosmf/resource-mgmt/rest/1.0/rdp/storage/dataset-attr/${_workflow-tenantID}/${_workflow-
templateName}/${_workflow-registryID}?size="MEDIUM"&type="VSAM"
<expectedStatusCode>200</expectedStatusCode>
<propertyMapping mapTo="DFH_ZOS_VSAM_VOLUME">["dataset-attributes-list"][0]["volser"]</propertyMapping>
</rest>

```

Figure 31. Sample request to get data set attributes

Create an LPAR resource pool entry

Use this operation to create an LPAR resource pool entry.

HTTP method and URI path

```
PUT /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<rdp-id>

Identifies the resource pool in which the LPAR entry is to be created.

Query parameters

None.

Description

This operation creates an LPAR resource pool entry in the specified LPAR resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the creation of an LPAR entry.

Request content

The request content is expected to contain a JSON object that describes the LPAR entry to be modified. See [Table 51 on page 79](#).

Table 51. Request content for the modify LPAR entry request

Field name	Type	Required or optional	Description
description	String	Optional	Description of the entry (up to 256 characters).
group	String	Optional	Group (up to 32 characters).
complete	Boolean	Optional	Indication of whether the LPAR entry is marked complete ("true" or "false"). An LPAR pool entry must be complete before it can be obtained for use. The default is false.
quiesced	Boolean	Optional	Indication of whether the LPAR entry is to be quiesced. The value is "true" if in use, otherwise "false." The default is false. An LPAR pool entry must not be quiesced for it to be obtained for use.
partition-name	String	Optional	Partition name (up to 8 characters).
partition-size	String	Optional	The size of the partition: "small", "medium", or "large".
group	String	Optional	Group name (up to 32 characters).
cpc-name	String	Optional	Central processing complex name (up to 8 characters).
sysname	String	Optional	System name (up to 8 characters).
os-config-name	String	Optional	OS config name (up to 8 characters).
zos-volumes-list	Array	Optional	Array of one or more objects that describe the z/OS volumes in the LPAR entry. See Table 52 on page 81 .
operational-volume	String	Optional	Operational volume (up to 6 characters).
operational-device	String	Optional	Operational device (up to 4 characters).
operational-zfs1-volume	String	Optional	Operational ZFS 1 volume (up to 6 characters).
operational-zfs1-device	String	Optional	Operational ZFS 1 device (up to 4 characters).
operational-zfs2-volume	String	Optional	Operational ZFS 2 volume (up to 6 characters).
operational-zfs2-device	String	Optional	Operational ZFS 2 device (up to 4 characters).
page-volume	String	Optional	Page volume (up to 6 characters).
page-device	String	Optional	Page device (up to 4 characters).
smf-volume	String	Optional	SMF volume (up to 6 characters).
smf-device	String	Optional	SMF device (up to 4 characters).

Table 51. Request content for the modify LPAR entry request (continued)

Field name	Type	Required or optional	Description
standalone-dump-volume	String	Optional	Stand-alone dump volume (up to 6 characters).
standalone-dump-device	String	Optional	Stand-alone dump device (up to 4 characters).
cds-primary-volume	String	Optional	Couple data sets primary volume (up to 6 characters).
cds-primary-device	String	Optional	Couple data sets primary device (up to 4 characters).
cds-alternate-volume	String	Optional	Couple data sets alternate volume (up to 6 characters).
cds-alternate-device	String	Optional	Couple data sets alternate device (up to 4 characters).
jes2-checkpoint1-volume	String	Optional	JES2 checkpoint 1 volume (up to 6 characters).
jes2-checkpoint1-device	String	Optional	JES2 checkpoint 1 device (up to 4 characters).
jes2-checkpoint2-volume	String	Optional	JES2 checkpoint 2 volume (up to 6 characters).
jes2-checkpoint2-device	String	Optional	JES2 checkpoint 2 device. (up to 4 characters).
jes2-spool1-volume	String	Optional	JES2 spool 1 volume (up to 6 characters).
jes2-spool1-device	String	Optional	JES2 spool 1 device. (up to 4 characters).
jes2-spool2-volume	String	Optional	JES2 spool 2 volume (up to 6 characters).
jes2-spool2-device	String	Optional	JES2 spool 2 device (up to 4 characters).
jes2-nje-node-list	Array of NJE nodes	Optional	The array of target Network Job Entry (NJE) nodes. See Table 53 on page 81 .
jes2-node-name	String	Optional	The JES2 local node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
jes2-ownnode	Integer	Optional	The JES2 node number assigned to the local node (a value from 1 to 32767).
ipv4-ip-address	String	Optional	IPv4 IP address (up to 15 characters).
ipv4-route	String	Optional	IPv4 route (up to 4 characters).
ipv4-default-route	String	Optional	IPv4 default route (up to 4 characters).

Table 51. Request content for the modify LPAR entry request (continued)

Field name	Type	Required or optional	Description
dns-domain-name	String	Optional	DNS domain name (up to 249 characters).
dns-hostname	String	Optional	DNS hostname (up to 8 characters).
tcpip-vlanid	String	Optional	TCPIP VLANID (a value from 1 to 4096). This field is valid with the installation of the PTF for APAR PH40058, which is available for z/OS V2R3, V2R4, and V2R5.
trle-name	String	Optional	Transport resource list element name.
trle-portname	String	Optional	Transport resource list element port name.
trle-device-read	String	Optional	Transport resource list element device read.
trle-device-write	String	Optional	Transport resource list element device write.
trle-device-datapath	String	Optional	Transport resource list element device data path.
sscp-id	String	Optional	SSCP identifier.
sms-system-volume	String	Optional	The volume for the SMS storage pool (up to 6 uppercase alphanumeric characters).
sms-system-device	String	Optional	The device for the SMS storage pool (up to 4 uppercase alphanumeric characters).
operational-sms1-volume	String	Optional	The volume for the SMS Active Control Data Set (ACDS) and Communications Data Set (COMMDS).
operational-sms1-device	String	Optional	The device for the SMS ACDS and COMMDS.
operational-sms2-volume	String	Optional	The volume for the SMS Source Control Data Set (SCDS).
operational-sms2-device	String	Optional	The device for the SMS SCDS.

Table 52. z/OS volumes object

Field name	Type	Description
volume	String	Volume serial number (up to 6 characters).
device	String	Device number (3 or 4 characters).

Table 53. NJE node

Field name	Type	Required or optional	Description
node-name	String	Required	The node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
ipaddr	String	Required	The IP address or fully qualified hostname of the node.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 54. HTTP error response codes for a create an LPAR resource pool entry request	
HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the LPAR resource pool entry. See [Table 55 on page 82](#).

Table 55. Response from a create an LPAR resource pool entry request		
Field	Type	Description
lpar-pool-id	String	The generated ID for new created LPAR entry.
object-uri	String	URI of the LPAR entry.

Modify an LPAR resource pool entry

Use this operation to modify the properties in an existing LPAR entry. An LPAR entry cannot be modified if it is in use.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/<lpar-pool-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<rdp-id>

Identifies the resource pool for the LPAR entry to be modified.

<lpar-pool-id>

Identifies the LPAR entry to be modified.

Query parameters

None.

Description

This operation modifies an LPAR resource pool entry in the specified LPAR resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the modification of an LPAR entry.

Request content

The request content is expected to contain a JSON object that describes the LPAR entry to be modified. See [Table 56 on page 83](#).

Table 56. Request content for the modify LPAR entry request			
Field name	Type	Required or optional	Description
description	String	Optional	Description of the entry (up to 32 characters).
complete	Boolean	Optional	Indication of whether the LPAR entry is complete (“true” or “false”).
quiesced	Boolean	Optional	Indication of whether the LPAR entry is quiesced. The value is “true” if in use, otherwise “false.” When quiesced, the LPAR entry cannot be obtained for use.
in-use	Boolean	Optional	Indication of whether the LPAR entry is in use (“true” or “false”).
partition-size	String	Optional.	The size of the partition: small, medium, or large.
partition-name	String	Optional	Partition name (up to 8 characters).
group	String	Optional	Group name (up to 32 characters).
cpc-name	String	Optional	Central processing complex name (up to 8 characters).
sysname	String	Optional	System name (up to 8 characters).
os-config-name	String	Optional	OS config name (up to 8 characters).
zos-volumes-list	Array	Optional	Array of one or more objects that describe the z/OS volumes in the LPAR entry. See Table 57 on page 85 .
operational-volume	String	Optional	Operational volume (up to 6 characters).
operational-device	String	Optional	Operational device (up to 4 characters).
operational-zfs1-volume	String	Optional	Operational ZFS 1 volume (up to 6 characters).
operational-zfs1-device	String	Optional	Operational ZFS 1 device (up to 4 characters).

Table 56. Request content for the modify LPAR entry request (continued)

Field name	Type	Required or optional	Description
operational-zfs2-volume	String	Optional	Operational ZFS 2 volume (up to 6 characters).
operational-zfs2-device	String	Optional	Operational ZFS 2 device (up to 4 characters).
page-volume	String	Optional	Page volume (up to 6 characters).
page-device	String	Optional	Page device (up to 4 characters).
smf-volume	String	Optional	SMF volume (up to 6 characters).
smf-device	String	Optional	SMF device (up to 4 characters).
standalone-dump-volume	String	Optional	Stand-alone dump volume (up to 6 characters).
standalone-dump-device	String	Optional	Stand-alone dump device (up to 4 characters).
cds-primary-volume	String	Optional	Couple data sets primary volume (up to 6 characters).
cds-primary-device	String	Optional	Couple data sets primary device (up to 4 characters).
cds-alternate-volume	String	Optional	Couple data sets alternate volume (up to 6 characters).
cds-alternate-device	String	Optional	Couple data sets alternate device (up to 4 characters).
jes2-checkpoint1-volume	String	Optional	JES2 checkpoint 1 volume (up to 6 characters).
jes2-checkpoint1-device	String	Optional	JES2 checkpoint 1 device (up to 4 characters).
jes2-checkpoint2-volume	String	Optional	JES2 checkpoint 2 volume (up to 6 characters).
jes2-checkpoint2-device	String	Optional	JES2 checkpoint 2 device. (up to 4 characters).
jes2-spool1-volume	String	Optional	JES2 spool 1 volume (up to 6 characters).
jes2-spool1-device	String	Optional	JES2 spool 1 device. (up to 4 characters).
jes2-spool2-volume	String	Optional	JES2 spool 2 volume (up to 6 characters).
jes2-spool2-device	String	Optional	JES2 spool 2 device (up to 4 characters).

Table 56. Request content for the modify LPAR entry request (continued)

Field name	Type	Required or optional	Description
jes2-nje-node-list	Array of NJE nodes	Optional	The array of target Network Job Entry (NJE) nodes. See Table 58 on page 86.
jes2-node-name	String	Optional	The JES2 local node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
jes2-ownnode	Integer	Optional	The JES2 node number assigned to the local node (a value from 1 to 32767).
ipv4-ip-address	String	Optional	IPv4 IP address (up to 15 characters).
ipv4-route	String	Optional	IPv4 route (up to 4 characters).
ipv4-default-route	String	Optional	IPv4 default route (up to 4 characters).
dns-domain-name	String	Optional	DNS domain name (up to 249 characters).
dns-hostname	String	Optional	DNS hostname (up to 8 characters).
tcpip-vlanid	String	Optional	TCPIP VLANID (a value from 1 to 4096). This field is valid with the PTF for APAR PH40058, which is available for z/OS V2R3, V2R4, and V2R5.
trle-name	String	Optional	Transport resource list element name.
trle-portname	String	Optional	Transport resource list element port name.
trle-device-read	String	Optional	Transport resource list element device read.
trle-device-write	String	Optional	Transport resource list element device write.
trle-device-datapath	String	Optional	Transport resource list element device data path.
sscp-id	String	Optional	SSCP identifier.
sms-system-volume	String	Optional	The volume for the SMS storage pool (up to 6 uppercase alphanumeric characters).
sms-system-device	String	Optional	The device for the SMS storage pool (up to 4 uppercase alphanumeric characters).
operational-sms1-volume	String	Optional	The volume for the SMS Active Control Data Set (ACDS) and Communications Data Set (COMMDS).
operational-sms1-device	String	Optional	The device for the SMS ACDS and COMMDS.
operational-sms2-volume	String	Optional	The volume for the SMS Source Control Data Set (SCDS).
operational-sms2-device	String	Optional	The device for the SMS SCDS.

Table 57. z/OS volumes object

Field name	Type	Description
volume	String	Volume serial number (up to 6 characters).

Table 57. z/OS volumes object (continued)

Field name	Type	Description
device	String	Device number (3- or 4-digit number).

Table 58. NJE node

Field name	Type	Required or optional	Description
node-name	String	Required	The node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
ipaddr	String	Required	The IP address or fully qualified hostname of the node.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 59. HTTP error response codes for a modify LPAR resource pool entry request

HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Delete an LPAR resource pool entry

Use this operation to delete an LPAR resource pool entry.

HTTP method and URI path

```
DELETE /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/<lpar-pool-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<rdp-id>

Identifies the resource pool in which the LPAR entry is to be deleted.

<lpar-pool-id>

Identifies the LPAR entry to be deleted.

Query parameters

None.

Description

This operation deletes an LPAR resource pool entry in the specified LPAR resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the deletion of an LPAR entry.

Request content

None.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 60. HTTP error response codes for a delete an LPAR resource pool entry request	
HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Retrieve a list of LPAR resource pool entries

Use this operation to retrieve a list of the LPAR entries in a specified resource pool.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/lpar/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<rdp-id>

Identifies the resource pool for which the LPAR entries are to be listed.

Query parameters

None.

Description

This operation retrieves a list of the LPAR entries in the specified resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the retrieval of a list of LPAR entries.

Request content

None.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102.](#)

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 61. HTTP error response codes for a list LPAR entries request</i>	
HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a list of one or more JSON objects, each containing details about an LPAR entry. See [Table 62 on page 88.](#)

<i>Table 62. Response from a list LPAR entries request</i>		
Field name	Type	Description
lpar-pool-id	String	Identifier of the LPAR entry.
object-uri	String	URI of the LPAR entry
description	String	Description of the entry (up to 32 characters).

Table 62. Response from a list LPAR entries request (continued)

Field name	Type	Description
complete	Boolean	Indication of whether the LPAR entry is complete (“true” or “false”).
quiesced	Boolean	Indication of whether the LPAR entry is quiesced. The value is “true” if in use, otherwise “false.” When quiesced, the LPAR entry cannot be obtained for use.
in-use	Boolean	Indication of whether the LPAR entry is in use (“true” or “false”).
instance-name	String	Name of the software services instance that obtained the pool entry.
rdp-name	String	Resource pool name.
rdp-id	String	Resource pool identifier.
partition-size	String	The size of the partition: small, medium, or large.
partition-name	String	Partition name (up to 8 characters).
group	String	Group name (up to 32 characters).
cpc-name	String	Central processing complex name (up to 8 characters).
sysname	String	System name (up to 8 characters).
os-config-name	String	OS config name (up to 8 characters).
zos-volumes-list	Array	Array of one or more objects that describe the z/OS volumes in the LPAR entry. See Table 63 on page 91 .
operational-volume	String	Operational volume (up to 6 characters).
operational-device	String	Operational device (up to 4 characters).
operational-zfs1-volume	String	Operational ZFS 1 volume (up to 6 characters).
operational-zfs1-device	String	Operational ZFS 1 device (up to 4 characters).
operational-zfs2-volume	String	Operational ZFS 2 volume (up to 6 characters).
operational-zfs2-device	String	Operational ZFS 2 device (up to 4 characters).
page-volume	String	Page volume (up to 6 characters).
page-device	String	Page device (up to 4 characters).
smf-volume	String	SMF volume (up to 6 characters).
smf-device	String	SMF device (up to 4 characters).
standalone-dump-volume	String	Stand-alone dump volume (up to 6 characters).
standalone-dump-device	String	Stand-alone dump device (up to 4 characters).
cds-primary-volume	String	Couple data sets primary volume (up to 6 characters).

Table 62. Response from a list LPAR entries request (continued)

Field name	Type	Description
cds-primary-device	String	Couple data sets primary device (up to 4 characters).
cds-alternate-volume	String	Couple data sets alternate volume (up to 6 characters).
cds-alternate-device	String	Couple data sets alternate device (up to 4 characters).
jes2-checkpoint1-volume	String	JES2 checkpoint 1 volume (up to 6 characters).
jes2-checkpoint1-device	String	JES2 checkpoint 1 device (up to 4 characters).
jes2-checkpoint2-volume	String	JES2 checkpoint 2 volume (up to 6 characters).
jes2-checkpoint2-device	String	JES2 checkpoint 2 device. (up to 4 characters).
jes2-spool1-volume	String	JES2 spool 1 volume (up to 6 characters).
jes2-spool1-device	String	JES2 spool 1 device. (up to 4 characters).
jes2-spool2-volume	String	JES2 spool 2 volume (up to 6 characters).
jes2-spool2-device	String	JES2 spool 2 device (up to 4 characters).
jes2-nje-node-list	Array of NJE nodes	The array of target Network Job Entry (NJE) nodes. See Table 64 on page 91 .
jes2-node-name	String	The JES2 local node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
jes2-ownnode	Integer	The JES2 node number assigned to the local node (a value from 1 to 32767).
ipv4-ip-address	String	IPv4 IP address (up to 15 characters).
ipv4-route	String	IPv4 route (up to 4 characters).
ipv4-default-route	String	IPv4 default route (up to 4 characters).
dns-domain-name	String	DNS domain name (up to 249 characters).
dns-hostname	String	DNS hostname (up to 8 characters).
trle-name	String	Transport resource list element name.
trle-portname	String	Transport resource list element port name.
trle-device-read	String	Transport resource list element device read.
trle-device-write	String	Transport resource list element device write.
trle-device-datapath	String	Transport resource list element device data path.
sscp-id	String	SSCP identifier.
sms-system-volume	String	The volume for the SMS storage pool (up to 6 uppercase alphanumeric characters).
sms-system-device	String	The device for the SMS storage pool (up to 4 uppercase alphanumeric characters).

Table 62. Response from a list LPAR entries request (continued)

Field name	Type	Description
operational-sms1-volume	String	The volume for the SMS Active Control Data Set (ACDS) and Communications Data Set (COMMDS).
operational-sms1-device	String	The device for the SMS ACDS and COMMDS.
operational-sms2-volume	String	The volume for the SMS Source Control Data Set (SCDS).
operational-sms2-device	String	The device for the SMS SCDS.

Table 63. z/OS volumes object

Field name	Type	Description
volume	String	Volume serial number (up to 6 characters).
device	String	Device number (3- or 4-digit number).

Table 64. NJE node

Field name	Type	Description
node-name	String	The node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
ipaddr	String	The IP address or fully qualified hostname of the node.

Retrieve the properties of an LPAR resource pool entry

Use this operation to retrieve the properties of an LPAR resource pool entry.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>rdp/<rdp-id>/lpar/<lpar-pool-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<rdp-id>

Identifies the resource pool that contains the LPAR entry.

<lpar-pool-id>

Identifies the LPAR entry to be modified.

Query parameters

None.

Description

This operation retrieves the properties of the specified LPAR entry in the specified resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the retrieval of the LPAR entry properties.

Request content

None.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 65. HTTP error response codes for a retrieve LPAR entry properties request	
HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains the properties for the specified LPAR entry. See [Table 66 on page 92](#).

Table 66. Response from a retrieve LPAR entry properties request		
Field name	Type	Description
lpar-pool-id	String	Identifier of the LPAR entry.
object-uri	String	URI of the LPAR entry
description	String	Description of the entry (up to 32 characters).
complete	Boolean	Indication of whether the LPAR entry is complete (“true” or “false”).
quiesced	Boolean	Indication of whether the LPAR entry is quiesced. The value is “true” if in use, otherwise “false.” When quiesced, the LPAR entry cannot be obtained for use.
in-use	Boolean	Indication of whether the LPAR entry is in use (“true” or “false”).
instance-name	String	Name of the software services instance that obtained the pool entry.
rdp-name	String	Resource pool name.

Table 66. Response from a retrieve LPAR entry properties request (continued)

Field name	Type	Description
rdp-id	String	Resource pool identifier.
partition-size	String	The size of the partition: small, medium, or large.
partition-name	String	Partition name (up to 8 characters).
group	String	Group name (up to 32 characters).
cpc-name	String	Central processing complex name (up to 8 characters).
sysname	String	System name (up to 8 characters).
os-config-name	String	OS config name (up to 8 characters).
zos-volumes-list	Array	Array of one or more objects that describe the z/OS volumes in the LPAR entry. See Table 67 on page 95 .
operational-volume	String	Operational volume (up to 6 characters).
operational-device	String	Operational device (up to 4 characters).
operational-zfs1-volume	String	Operational ZFS 1 volume (up to 6 characters).
operational-zfs1-device	String	Operational ZFS 1 device (up to 4 characters).
operational-zfs2-volume	String	Operational ZFS 2 volume (up to 6 characters).
operational-zfs2-device	String	Operational ZFS 2 device (up to 4 characters).
page-volume	String	Page volume (up to 6 characters).
page-device	String	Page device (up to 4 characters).
smf-volume	String	SMF volume (up to 6 characters).
smf-device	String	SMF device (up to 4 characters).
standalone-dump-volume	String	Stand-alone dump volume (up to 6 characters).
standalone-dump-device	String	Stand-alone dump device (up to 4 characters).
cds-primary-volume	String	Couple data sets primary volume (up to 6 characters).
cds-primary-device	String	Couple data sets primary device (up to 4 characters).
cds-alternate-volume	String	Couple data sets alternate volume (up to 6 characters).
cds-alternate-device	String	Couple data sets alternate device (up to 4 characters).
jes2-checkpoint1-volume	String	JES2 checkpoint 1 volume (up to 6 characters).
jes2-checkpoint1-device	String	JES2 checkpoint 1 device (up to 4 characters).
jes2-checkpoint2-volume	String	JES2 checkpoint 2 volume (up to 6 characters).

Table 66. Response from a retrieve LPAR entry properties request (continued)

Field name	Type	Description
jes2-checkpoint2-device	String	JES2 checkpoint 2 device. (up to 4 characters).
jes2-spool1-volume	String	JES2 spool 1 volume (up to 6 characters).
jes2-spool1-device	String	JES2 spool 1 device. (up to 4 characters).
jes2-spool2-volume	String	JES2 spool 2 volume (up to 6 characters).
jes2-spool2-device	String	JES2 spool 2 device (up to 4 characters).
jes2-nje-node-list	Array of NJE nodes	The array of target Network Job Entry (NJE) nodes. Sees Table 68 on page 95 .
jes2-node-name	String	The JES2 local node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
jes2-ownnode	Integer	The JES2 node number assigned to the local node (a value from 1 to 32767).
ipv4-ip-address	String	IPv4 IP address (up to 15 characters).
ipv4-route	String	IPv4 route (up to 4 characters).
ipv4-default-route	String	IPv4 default route (up to 4 characters).
dns-domain-name	String	DNS domain name (up to 249 characters).
dns-hostname	String	DNS hostname (up to 8 characters).
tcpip-vlanid	String	TCPIP VLANID (a value from 1 to 4096). This field is valid with the PTF for APAR PH40058, which is available for z/OS V2R3, V2R4, and V2R5.
trle-name	String	Transport resource list element name.
trle-portname	String	Transport resource list element port name.
trle-device-read	String	Transport resource list element device read.
trle-device-write	String	Transport resource list element device write.
trle-device-datapath	String	Transport resource list element device data path.
sscp-id	String	SSCP identifier.
sms-system-volume	String	The volume for the SMS storage pool (up to 6 uppercase alphanumeric characters).
sms-system-device	String	The device for the SMS storage pool (up to 4 uppercase alphanumeric characters).
operational-sms1-volume	String	The volume for the SMS Active Control Data Set (ACDS) and Communications Data Set (COMMDS).
operational-sms1-device	String	The device for the SMS ACDS and COMMDS.
operational-sms2-volume	String	The volume for the SMS Source Control Data Set (SCDS).
operational-sms2-device	String	The device for the SMS SCDS.

Table 67. z/OS volumes object

Field name	Type	Description
volume	String	Volume serial number (up to 6 characters).
device	String	Device number (3- or 4-digit number).

Table 68. NJE node

Field name	Type	Description
node-name	String	The node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
ipaddr	String	The IP address or fully qualified hostname of the node.

Obtain an LPAR resource pool entry

Use this operation to obtain an LPAR entry for use.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/lpar/actions/obtain
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation obtains an LPAR entry from the LPAR resource pool.

Note: An LPAR entry in the quiesced state cannot be obtained for use.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the modification of an LPAR entry.

Request content

The request content is expected to contain a JSON object that describes the LPAR entry to be obtained. See [Table 69 on page 95](#).

Table 69. Request content for the obtain LPAR entry request

Field name	Type	Required or optional	Description
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \$_workflow-templateID.

Table 69. Request content for the obtain LPAR entry request (continued)

Field name	Type	Required or optional	Description
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
registry-uuid	String	See description.	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> Provisioning workflows: \${_workflow-registryID} Action workflows: \${_workflow-parentRegistryID}
lpar-parms	JSON object	Required	LPAR parameters for the request. See Table 70 on page 96 .

Table 70. LPAR parameter fields

Field	Type	Required or optional	Description
name	String	Required	Name of the software services instance. Use the workflow internal variable \${_workflow-softwareServiceInstanceName} for this parameter.
cpc-name	String	Optional	Central processing complex name to be used for the selection of an LPAR entry. Up to 8 characters can be specified.
partition-size	String	Required	The size of the partition to be obtained: Small, medium, large, or any.
group	String	Optional	Group name (up to 32 characters).

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 71. HTTP error response codes for an obtain LPAR resource pool entry request

HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.

Table 71. HTTP error response codes for an obtain LPAR resource pool entry request (continued)

HTTP error status code	Description
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains details about the LPAR entry that was obtained. See [Table 72 on page 97](#).

Table 72. Response from an obtain LPAR entry properties request

Field name	Type	Description
lpar-pool-id	String	Identifier of the LPAR entry.
object-uri	String	URI of the LPAR entry
description	String	Description of the entry (up to 32 characters).
complete	Boolean	Indication of whether the LPAR entry is complete (“true” or “false”).
quiesced	Boolean	Indication of whether the LPAR entry is quiesced. The value is “true” if in use, otherwise “false.” When quiesced, the LPAR entry cannot be obtained for use.
in-use	Boolean	Indication of whether the LPAR entry is in use (“true” or “false”).
instance-name	String	Name of the software services instance that obtained the pool entry.
rdp-name	String	Resource pool name.
rdp-id	String	Resource pool identifier.
rdp-job-statement	String	The JOB statement from the resource pool.
partition-size	String	The size of the partition: small, medium, or large.
partition-name	String	Partition name (up to 8 characters).
group	String	Group name (up to 32 characters).
cpc-name	String	Central processing complex name (up to 8 characters).
sysname	String	System name (up to 8 characters).
os-config-name	String	OS config name (up to 8 characters).
zos-volumes-list	Array	Array of one or more objects that describe the z/OS volumes in the LPAR entry. See Table 73 on page 99 .
operational-volume	String	Operational volume (up to 6 characters).
operational-device	String	Operational device (up to 4 characters).
operational-zfs1-volume	String	Operational ZFS 1 volume (up to 6 characters).
operational-zfs1-device	String	Operational ZFS 1 device (up to 4 characters).

Table 72. Response from an obtain LPAR entry properties request (continued)

Field name	Type	Description
operational-zfs2-volume	String	Operational ZFS 2 volume (up to 6 characters).
operational-zfs2-device	String	Operational ZFS 2 device (up to 4 characters).
page-volume	String	Page volume (up to 6 characters).
page-device	String	Page device (up to 4 characters).
smf-volume	String	SMF volume (up to 6 characters).
smf-device	String	SMF device (up to 4 characters).
standalone-dump-volume	String	Stand-alone dump volume (up to 6 characters).
standalone-dump-device	String	Stand-alone dump device (up to 4 characters).
cds-primary-volume	String	Couple data sets primary volume (up to 6 characters).
cds-primary-device	String	Couple data sets primary device (up to 4 characters).
cds-alternate-volume	String	Couple data sets alternate volume (up to 6 characters).
cds-alternate-device	String	Couple data sets alternate device (up to 4 characters).
jes2-checkpoint1-volume	String	JES2 checkpoint 1 volume (up to 6 characters).
jes2-checkpoint1-device	String	JES2 checkpoint 1 device (up to 4 characters).
jes2-checkpoint2-volume	String	JES2 checkpoint 2 volume (up to 6 characters).
jes2-checkpoint2-device	String	JES2 checkpoint 2 device. (up to 4 characters).
jes2-spool1-volume	String	JES2 spool 1 volume (up to 6 characters).
jes2-spool1-device	String	JES2 spool 1 device. (up to 4 characters).
jes2-spool2-volume	String	JES2 spool 2 volume (up to 6 characters).
jes2-spool2-device	String	JES2 spool 2 device (up to 4 characters).
jes2-nje-node-list	Array of NJE nodes	The array of target Network Job Entry (NJE) nodes. See Table 74 on page 99 .
jes2-node-name	String	The JES2 local node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
jes2-ownnode	Integer	The JES2 node number assigned to the local node (a value from 1 to 32767).
ipv4-ip-address	String	IPv4 IP address (up to 15 characters).
ipv4-route	String	IPv4 route (up to 4 characters).
ipv4-default-route	String	IPv4 default route (up to 4 characters).
dns-domain-name	String	DNS domain name (up to 249 characters).

Table 72. Response from an obtain LPAR entry properties request (continued)

Field name	Type	Description
dns-hostname	String	DNS hostname (up to 8 characters).
tcpip-vlanid	String	TCPIP VLANID (a value from 1 to 4096). This field is valid with the PTF for APAR PH40058, which is available for z/OS V2R3, V2R4, and V2R5.
trle-name	String	Transport resource list element name.
trle-portname	String	Transport resource list element port name.
trle-device-read	String	Transport resource list element device read.
trle-device-write	String	Transport resource list element device write.
trle-device-datapath	String	Transport resource list element device data path.
sscp-id	String	SSCP identifier.
sms-system-volume	String	The volume for the SMS storage pool (up to 6 uppercase alphanumeric characters).
sms-system-device	String	The device for the SMS storage pool (up to 4 uppercase alphanumeric characters).
operational-sms1-volume	String	The volume for the SMS Active Control Data Set (ACDS) and Communications Data Set (COMMDS).
operational-sms1-device	String	The device for the SMS ACDS and COMMDS.
operational-sms2-volume	String	The volume for the SMS Source Control Data Set (SCDS).
operational-sms2-device	String	The device for the SMS SCDS.

Table 73. z/OS volumes object

Field name	Type	Description
volume	String	Volume serial number (up to 6 characters).
device	String	Device number (3- or 4-digit number).

Table 74. NJE node

Field name	Type	Description
node-name	String	The node name (up to 8 uppercase alphanumeric or special (\$, #, or @) characters).
ipaddr	String	The IP address or fully qualified hostname of the node.

Example HTTP interaction

In [Figure 32 on page 100](#), a request is submitted to obtain an LPAR entry.

```

POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/lpar/obtain
{
  "registry-uuid": "${_workflow-registryID}",
  "template-name": "${_workflow-templateName}",
  "tenant-id": "${_workflow-tenantID}",
  "lpar-parms": {
    "partition-size": "any",
    "cpc-name": "cpc1",
    "group": "sales",
    "name": "${_workflow-softwareServiceInstanceName}"
  }
}

```

Figure 32. Sample request to obtain an LPAR entry

Release an LPAR resource pool entry

Use this operation to release an LPAR entry when it is no longer needed.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/lpar/actions/release
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

Query parameters

None.

Description

This operation releases an LPAR entry from the LPAR resource pool. A released entry is available for use.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the modification of an LPAR entry.

Request content

The request content is expected to contain a JSON object that describes the LPAR entry to be released. See [Table 75 on page 100](#).

Table 75. Request content for the release LPAR entry request

Field name	Type	Required or optional	Description
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.

Table 75. Request content for the release LPAR entry request (continued)

Field name	Type	Required or optional	Description
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
registry-uid	String	See description.	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> Provisioning workflows: \${_workflow-registryID} Action workflows: \${_workflow-parentRegistryID}
lpar-parms	JSON object	Required	LPAR parameters for the request. See Table 76 on page 101 .

Table 76. LPAR parameter fields

Field	Type	Required or optional	Description
lpar-pool-id	String	Required	Identifier of the LPAR entry.

Authorization requirements

The user must be a domain administrator for the domain that contains the LPAR resource pool.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 77. HTTP error response codes for a release LPAR resource pool entry request

HTTP error status code	Description
HTTP 400 Bad request	The request cannot be processed because it contains a syntax error or an incorrect parameter.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

None.

Example HTTP interaction

In [Figure 33 on page 102](#), a request is submitted to release an LPAR entry.

```

POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/lpar/release
{
  "template-name": "${_workflow-templateName}",
  "tenant-id": "${_workflow-tenantID}",
  "lpar-params": {
    "lpar-pool-id": "IYU1000000"
  }
}

```

Figure 33. Sample request to release an LPAR entry

Resource management services

The resource management services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to get and list domains, tenants, and resource pools that were defined in support of IBM Cloud Provisioning and Management for z/OS.

Table 78 on page 102 lists the operations that the resource management services provide.

Resource management services

Table 78. z/OSMF resource management services: operations summary

Operation name	HTTP method and URI path
“Get a domain” on page 104	GET /zosmf/resource-mgmt/rest/<version>/domains/<object-id>
“Get a domain history” on page 110	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/history
“List the domains” on page 112	GET /zosmf/resource-mgmt/rest/<version>/domains/
“Create a tenant” on page 117	PUT /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/tenants
“Get a tenant” on page 121	GET /zosmf/resource-mgmt/rest/<version>/tenants/<object-id>
“Get a tenant history” on page 126	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/history
“List the tenants” on page 129	GET /zosmf/resource-mgmt/rest/<version>/tenants/
“Delete a tenant” on page 135	DELETE /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>
“Assign CPU properties to a tenant” on page 136	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-cpu-capping-properties
“Assign memory capping properties to a tenant” on page 138	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-memory-capping-properties
“Assign a solution ID” on page 140	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-solution-id

Table 78. z/OSMF resource management services: operations summary (continued)	
Operation name	HTTP method and URI path
“Disable CPU capping” on page 141	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-cpu-capping
“Disable memory capping” on page 143	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-memory-capping
“Disable metering” on page 144	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-metering
“Enable CPU capping” on page 146	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-cpu-capping
“Enable memory capping” on page 147	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-memory-capping
“Enable metering” on page 149	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-metering
“Add tenant consumer” on page 150	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/add
“Remove tenant consumer” on page 152	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/remove
“Add tenant description” on page 153	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/description/actions/add
“Add tenant groups” on page 155	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/add
“Remove tenant groups” on page 156	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/remove
“Get a resource pool” on page 158	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/<rdp-id>
“Get a domain resource pool” on page 164	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/<rdp-id>
“Get a resource pool history” on page 171	GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/history
“List the resource pools” on page 173	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/
“List domain resource pools” on page 179	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/

Table 78. z/OSMF resource management services: operations summary (continued)

Operation name	HTTP method and URI path
“List template resource pools” on page 186	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/template/<template-name>/rdp
“Update the security state for a tenant” on page 192	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/state/actions/update
“Get security resources” on page 194	GET /zosmf/resource-mgmt/rest/<version>/security-resources

Authorization requirements

Use of the Resource Management services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the provisioning administrator and domain administrator. The specific requirements for each resource management service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 49](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 400 Bad request

There is a missing field in the request body.

HTTP 401 Not authorized

The request cannot be processed because the client is not authorized.

HTTP 403 Cannot access

The client does not have access rights to the content (they are not authorized). As a result, the server is not returning the expected.

HTTP 404 Not found

The requested resource does not exist.

HTTP 409 Conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

HTTP 500 Server error

The server encountered an error when it processed the request.

Get a domain

Use this operation to retrieve a domain.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<object-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<object-id>

Identifies the domain to be retrieved.

Query parameters

None.

Description

This operation retrieves a domain.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a domain being retrieved.

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 105](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 79. HTTP error response codes for a get domain request	
HTTP error status code	Description
HTTP 400 Bad request	The request contains incorrect parameters.
HTTP 404 Not found	The requested domain does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the domain. See [Table 80 on page 105](#).

Table 80. Response from a get domain request		
Field	Type	Description
domain-id	String	The generated ID for the domain.
domain-name	String	Descriptive name for the domain.

Table 80. Response from a get domain request (continued)

Field	Type	Description
domain-system-list	Array	Array describing the systems in the domain. See Table 81 on page 108 .
domain-administrator-list	Array of Strings	List of user IDs for the domain administrators.
domain-administrator-group-list	Array of Strings	List of SAF groups for the domain administrators.
network-administrator-list	Array of Strings	List of user IDs for the network administrators.
network-administrator-group-list	Array of Strings	List of SAF groups for the network administrators.
wlm-administrator-list	Array of Strings	List of user IDs for the WLM administrators.
wlm-administrator-group-list	Array of Strings	List of SAF groups for the WLM administrators.
security-administrator	String	User ID of the security administrator.
security-job-statement	String	JOB statement JCL used in security jobs for the domain.
domain-approver-list	Array of Strings	List of user IDs for the domain approvers.
domain-approver-group-list	Array of Strings	List of SAF groups for the domain approvers.
object-uri	String	URI of the newly created object.
domain-description	String	Description of the domain.
automatic-security	boolean	Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization: <ul style="list-style-type: none"> • true if the domain is set up for automatic authorization • false if the domain is setup for manual authorization.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 82 on page 109 .
local-system	JSON object	System object for the local system. See Table 81 on page 108 .

Table 80. Response from a get domain request (continued)

Field	Type	Description
domain-state	String	<p>State of the domain:</p> <p>network_update_failed Indicates that an attempt to modify the network cloud domain that is associated with the domain failed. The network cloud domain is modified when you modify the domain's network administrators or the domain's systems.</p> <p>security_update_failed Indicates that the security workflow that provides automatic security failed.</p> <p>pending_security_update Indicates one of the following:</p> <ul style="list-style-type: none"> • Manual Security definition was selected for the domain, and security setup is required. • Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. <p>Operational Indicates that the domain is ready for use.</p>
security-job-disposition	String	<p>Disposition of security jobs after they complete successfully:</p> <p>manual The domain is configured for manual security.</p> <p>keep Keep jobs on completion. Jobs that are dynamically submitted for security are kept after they complete. This is the default. You can manually delete the jobs.</p> <p>delete Delete jobs on completion. Jobs that are dynamically submitted for security are deleted automatically after they complete.</p>

Table 80. Response from a get domain request (continued)

Field	Type	Description
security-workflow-disposition	String	Disposition of security workflows after they complete successfully: manual The domain is configured for manual security. delete Delete successful workflows on completion. Workflows that are used for security are deleted automatically after they complete successfully. This is the default. keep Keep successful workflows on completion. Workflows that are used for security are kept after they complete successfully. You can manually delete the workflows using the Workflows table in the Workflows task.
create-time	String	Date and time that the domain was created.
created-by-user	String	User who created the domain.
last-modified-time	String	The date and time of the last modification to the domain.
last-modified-by-user	String	User who last modified the domain.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.
domain-state-error-details	String	If present, contains details about the domain's error state.
domain-shared-rdp-id	String	The ID of the shared resource pool in this domain. It is empty until a shared resource pool in the domain is created.
history-archive-directory	String	The directory path leading to archived history entries. If no archive directory path is specified, archiving is not enabled and an empty string will be returned.

Table 81. Response from a get request: Systems

Field	Type	Description
sysplex-name	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
sysplex-node-name	String	Sysplex node name.

Table 81. Response from a get request: Systems (continued)

Field	Type	Description
system-nickname	String	Unique name that is assigned to the system definition.

Table 82. Response from a create request: SAF-resource object

Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Example HTTP interaction

In Figure 34 on page 109, a request is submitted to retrieve a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/<object-id>
```

Figure 34. Sample request to get a domain

The following is the response body for the example get domain request.

```
{
  "domain-id": "IYU1",
  "domain-name": "Domain1",
  "domain-state": "operational",
  "domain-system-list": [{
    "sysplex-name": "PLEX1",
    "sysplex-node-name": "SYS1",
    "system-nickname": "SYS1"
  }],
  "domain-administrator-list": ["PROVADM"],
  "domain-administrator-group-list": ["DAGRP1", "DAGRP2", ...],
  "network-administrator-list": ["netadmin"],
  "network-administrator-group-list": ["NAGRP1", ...],
  "security-administrator": "secadmin",
  "security-job-statement": "//JOB CARD JOB (ACCTINFO)",
}
```

```

"automatic-security": true,
"wlm-administrator-list": ["wlmadmin"],
"wlm-administrator-group-list": ["WAGRP1",...],
"domain-approver-list": ["approver"],
"domain-approver-group-list": ["AAGRP1",...],
"object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1",
"domain-description": "domain description",
"create-time": "2017-11-15T19:12:45.723Z",
"created-by-user": "PROVADM",
"last-modified-time": "2017-11-15T19:13:17.451Z",
"last-modified-by-user": "PROVADM",
"local-system": {
  "sysplex-name": "PLEX1",
  "sysplex-node-name": "SY1",
  "system-nickname": "SY1"
},
"SAF-resources": [{
  "description": "Designates the user as a z/OSMF user with authorization to log in.",
  "ids": ["PROVADM"],
  "groups": [],
  "role": "Domain Administrator",
  "resource-class": "ZMFAPLA",
  "resource-name": "IZUDFLT.ZOSMF",
  "required-access": "SAF_READ",
  "other-required-ids": [],
  "audit-requirements": ""
}, ... ],
"security-job-disposition": "delete",
"security-workflow-disposition": "delete",
"provisioning-version": "1200",
"provisioning-version-supported": true,
"domain-shared-rdp-id": "IYU1ZZZZ"
}

```

Get a domain history

Use this operation to retrieve a domain history.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/history
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<domain-id>

Identifies the domain for which history is to be retrieved.

Query parameters

None.

Description

This operation retrieves the history for a domain.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 111](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 111](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 83. HTTP error response codes for a get domain history request</i>	
HTTP error status code	Description
HTTP 400 Bad request	The request contains incorrect parameters.
HTTP 404 Not found	The requested domain does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the domain.

[Table 84 on page 111](#) lists the fields in the history object.

<i>Table 84. Response from a get request: History object</i>		
Field	Type	Description
action-type	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none">• Create• Add administrator• Add approver• Add system• Create pool• Create tenant• Delete pool• Delete tenant• Remove administrator• Remove approver• Remove system• Update description• Update state• Update history setting
user	String	The user who performed the action.
action-time	String	The time that the action was taken.

Table 84. Response from a get request: History object (continued)

Field	Type	Description
action-details	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

Example HTTP interaction

In Figure 35 on page 112, a request is submitted to retrieve the history for a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/history
```

Figure 35. Sample request to get a domain history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:28:38.133Z",
      "action-details": "Created domain"
    },
    {
      "action-type": "Update description",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:28:55.880Z",
      "action-details": "Updated domain description"
    }
  ]
}
```

List the domains

Use this operation to list the domains that are defined for IBM Cloud Provisioning and Management for z/OS.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation lists the domains for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 113](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 85. HTTP error response codes for a list domains request</i>	
HTTP error status code	Description
HTTP 400 Bad request	The request contains incorrect parameters.
HTTP 404 Not found	The requested domain does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the domains. See [Table 86 on page 113](#).

<i>Table 86. Response from a list domains request</i>		
Field	Type	Description
domain-list	Array	Domains. See Table 87 on page 113 .
local-system	JSON object	System object for the local system. See Table 81 on page 108 .

<i>Table 87. Properties of domains</i>		
Field	Type	Description
domain-id	String	The generated ID for the domain.
domain-name	String	Descriptive name for the domain.

Table 87. Properties of domains (continued)

Field	Type	Description
domain-state	String	<p>State of the domain:</p> <p>network_update_failed Indicates that an attempt to modify the network cloud domain that is associated with the domain failed. The network cloud domain is modified when you modify the domain's network administrators or the domain's systems.</p> <p>security_update_failed Indicates that the security workflow that provides automatic security failed.</p> <p>pending_security_update Indicates one of the following:</p> <ul style="list-style-type: none"> • Manual Security definition was selected for the domain, and security setup is required. • Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. <p>Operational Indicates that the domain is ready for use.</p>
domain-system-list	Array	Array describing the systems in the domain. See Table 81 on page 108 .
domain-administrator-list	Array of Strings	List of user IDs for the domain administrators.
network-administrator-list	Array of Strings	List of user IDs for the network administrators.
security-administrator	String	User ID of the security administrator.
security-job-statement	String	JOB statement JCL used in security jobs for the domain.
automatic-security	boolean	<p>Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization:</p> <ul style="list-style-type: none"> • true if the domain is set up for automatic authorization • false if the domain is setup for manual authorization.
wlm-administrator-list	Array of Strings	List of user IDs for the WLM administrators.
domain-approver-list	Array of Strings	List of user IDs for the domain approvers.
object-uri	String	URI of the newly created object.
domain-description	String	Description of the domain.
create-time	String	Date and time that the domain was created.
created-by-user	String	User who created the domain.

Table 87. Properties of domains (continued)

Field	Type	Description
last-modified-time	String	The date and time of the last modification to the domain.
last-modified-by-user	String	User who last modified the domain.
security-job-disposition	String	Disposition of security jobs after they complete successfully: manual The domain is configured for manual security. keep Keep jobs on completion. Jobs that are dynamically submitted for security are kept after they complete. This is the default. You can manually delete the jobs. delete Delete jobs on completion. Jobs that are dynamically submitted for security are deleted automatically after they complete.
security-workflow-disposition	String	Disposition of security workflows after they complete successfully: manual The domain is configured for manual security. delete Delete successful workflows on completion. Workflows that are used for security are deleted automatically after they complete successfully. This is the default. keep Keep successful workflows on completion. Workflows that are used for security are kept after they complete successfully. You can manually delete the workflows using the Workflows table in the Workflows task.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 82 on page 109 .
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.
domain-state-error-details	String	If present, contains details about the domain's error state.
domain-shared-rdp-id	String	The ID of the shared resource pool in this domain. It is empty until a shared resource pool in the domain is created.

Example HTTP interaction

In Figure 36 on page 116, a request is submitted to list the domains.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/
```

Figure 36. Sample request to list domains

The following is the response body for the example list domains request.

```
{
  "domain-list": [{
    "domain-id": "izu$0",
    "domain-name": "default",
    "domain-system-list": [{
      "sysplex-name": "DUMBPLEX",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE_001"},
    ...
  ],
  "domain-administrator-list": ["ZOSMFT1", ... ],
  "network-administrator-list": ["ZOSMFT1", ... ],
  "wlm-administrator-list": ["ZOSMFT1", ... ],
  "security-administrator": "ZOSMFT1",
  "security-job-statement": "//JOB CARD JOB(acct-info)",
  "domain-approver-list": ["ZOSMFT1", ... ],
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/izu$0",
  "domain-description": "default domain"
},
...
]
```

```
{
  "domain-list": [{
    "domain-id": "IYU0",
    "domain-name": "default",
    "domain-state": "operational",
    "domain-system-list": [{
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SYS1",
      "system-nickname": "SYS1"
    }],
    "domain-administrator-list": [],
    "network-administrator-list": [],
    "security-administrator": null,
    "security-job-statement": "",
    "automatic-security": true,
    "wlm-administrator-list": [],
    "domain-approver-list": [],
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU0",
    "domain-description": "default domain",
    "create-time": "2016-10-19T08:09:08.648Z",
    "created-by-user": "izusvr",
    "last-modified-time": "2016-10-19T08:09:08.648Z",
    "last-modified-by-user": "izusvr",
    "security-job-disposition": "keep",
    "security-workflow-disposition": "delete",
    "provisioning-version": "1200",
    "provisioning-version-supported": true,
    "domain-shared-rdp-id": "IYU0ZZZZ"
  }, {
    "domain-id": "IYU1",
    "domain-name": "Domain1",
    "domain-state": "operational",
    "domain-system-list": [{
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SYS1",
      "system-nickname": "SYS1"
    }],
    "domain-administrator-list": ["PROVADM"],
    "network-administrator-list": ["netadmin"],
    "security-administrator": "secadmin",
    "security-job-statement": "//JOB CARD JOB (ACCTINFO)",
    "automatic-security": true,

```

```

    "wlm-administrator-list": ["wladmin"],
    "domain-approver-list": ["approver"],
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1",
    "domain-description": "domain description",
    "create-time": "2017-10-15T19:12:45.723Z",
    "created-by-user": "PROVADM",
    "last-modified-time": "2017-10-15T19:13:17.451Z",
    "last-modified-by-user": "PROVADM",
    "SAF-resources": [{
      "description": "Designates the user as a z/OSMF user with authorization to log in.",
      "ids": ["PROVADM"],
      "groups": [],
      "role": "Domain Administrator",
      "resource-class": "ZMFAPLA",
      "resource-name": "IZUDFLT.ZOSMF",
      "required-access": "SAF_READ",
      "other-required-ids": [],
      "audit-requirements": ""
    }, ... ],
    "security-job-disposition": "delete",
    "security-workflow-disposition": "delete",
    "provisioning-version": "1200",
    "provisioning-version-supported": true,
    "domain-shared-rdp-id": "IYU0ZZZZ"
  }],
  "local-system": {
    "sysplex-name": "PLEX1",
    "sysplex-node-name": "SYS1",
    "system-nickname": "SYS1"
  }
}

```

Create a tenant

Use this operation to create a tenant.

HTTP method and URI path

```
PUT /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/tenants
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<domain-id>

Identifies the domain in which to create the tenant.

Query parameters

None.

Description

This operation creates a tenant on the specified domain.

On successful completion, HTTP status code 201 is returned, indicating that the request resulted in a tenant being created.

Request content

The request content is expected to contain a JSON object that describes the tenant to be created. See [Table 88 on page 118](#).

Table 88. Request content for the create tenant request

Field name	Type	Required or optional	Description
tenant-name	String	Required	Name of the tenant.
tenant-description	String	Optional	Description of the tenant.
tenant-consumer-list	Array	Optional	List of consumer user IDs for the tenant.
tenant-group-list	Array	Optional	List of groups in the tenant.
tenant-metering-capping-properties	JSON object	Optional	Object that describes the metering and capping properties to set for the tenant. See Table 89 on page 118 .
tenant-managed-by	JSON object	Optional	Object that describes the tenant managed by information. See Table 90 on page 119 .

Table 89. Tenant metering and capping properties object

Field	Type	Required or optional	Description
tenant-capping-enabled	Boolean	Optional	If true, enables CPU capping on the specified tenant.
tenant-cpu-cap-limit	Integer	Optional	Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type: lpar-share-percentage 1-100. service-unit 1-999999999 cp 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP. msu 1-999999999.

Table 89. Tenant metering and capping properties object (continued)

Field	Type	Required or optional	Description
tenant-cpu-cap-type	String	Optional	<p>Indicates the type of capping for the tenant. Values are:</p> <p>lpar-share-percentage Percentage of the LPAR share in the general purpose processor pool.</p> <p>service-unit Unweighted CPU service units per second.</p> <p>cp A number of general purpose processors (CPs), including numbers with up to two decimal places.</p> <p>msu Millions of service units per hour.</p> <p>none Removes all of the capping properties.</p>
tenant-memory-cap-limit	Integer	Optional	The limit in gigabytes of the memory cap.
tenant-memory-capping-enabled	Boolean	Optional	If true, enables memory capping on the specified tenant.
tenant-metering-enabled	Boolean	Optional	If true, enables metering on the specified tenant.
tenant-solution-id	String	Optional	Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.

Table 90. Tenant managed by request object

Field	Type	Required or optional	Description
system	JSON Object	Optional	Object that describes the managing system. See Table 91 on page 120 .
tenant-id	String	Required	The generated ID for the managing tenant.
tenant-name	String	Required	Descriptive name for the managing tenant.
zosmf-url	String	Required	The URL used to access the managing z/OSMF instance.

Table 91. Systems request object

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex. The name is the value that is specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
sysplex-node-name	String	Required	Sysplex node name.
system-nickname	String	Required	Unique name that is assigned to the system definition.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 201 is returned, and with a response body. Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 92. HTTP error response codes for a get domain request

HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 401 Not authorized	The request cannot be processed because the client is not authorized.
HTTP 403 Cannot access	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404 Not found	The requested resource does not exist.
HTTP 409 Conflict	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the tenant. See [Table 93 on page 120](#).

Table 93. Response from a create tenant request

Field	Type	Description
tenant-id	String	The generated ID for the tenant.
object-uri	String	URI of the tenant.
tenant-domain-id	String	The generated ID for the domain to which the tenant belongs.

Example HTTP interaction

In Figure 37 on page 121, a request is submitted to retrieve a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/tenants
```

```
{
  "tenant-name": "string",
  "tenant-description": "string",
  "tenant-consumer-list": [
    "string"
  ],
  "tenant-group-list": [
    "string"
  ],
  "tenant-metering-capping-properties": {
    "tenant-capping-enabled": true,
    "tenant-cpu-cap-limit": 0,
    "tenant-cpu-cap-type": "lpar-share-percentage",
    "tenant-memory-cap-limit": 0,
    "tenant-memory-capping-enabled": true,
    "tenant-metering-enabled": true,
    "tenant-solution-id": "string"
  },
  "tenant-managed-by": {
    "system": {
      "sysplex-name": "string",
      "sysplex-node-name": "string",
      "system-nickname": "string"
    },
    "tenant-id": "string",
    "tenant-name": "string",
    "zosmf-url": "string"
  }
}
```

Figure 37. Sample request to create a tenant

The following is the response body for the example create tenant request.

```
{
  "tenant-id": "string",
  "object-uri": "string",
  "tenant-domain-id": "string"
}
```

Get a tenant

Use this operation to retrieve a tenant.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<object-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<object-id>

Identifies the tenant to be retrieved.

Query parameters

None.

Description

This operation retrieves a tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a tenant being retrieved, and a response body is returned. See [“Response content” on page 122](#).

Request content

None.

Authorization requirements

The user must be a domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 94. HTTP error response codes for a get tenant request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested tenant does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the tenant. See [Table 95 on page 122](#).

Table 95. Response from a get tenant request		
Field	Type	Description
tenant-id	String	Generated ID for the tenant.
tenant-name	String	Descriptive name for the tenant.
tenant-shared-rdp-id	String	The ID of the shared resource pool that is associated with this tenant. It is empty until a shared resource pool is created. The resource pool ID suffix ZZ is reserved for the shared resource pool of the tenant. Dedicated resource pools cannot have ZZ as an ID suffix.
tenant-domain-id	String	Generated ID for the domain to which the tenant belongs.
tenant-domain-name	String	Descriptive name of the domain to which the tenant belongs.
tenant-templates	Array of Strings	Array that describes the templates that are associated with dedicated resource pools. See Table 96 on page 125 . For templates that are associated with shared resource pools, see the rdp-shared-template-name-list field in a Get resource pool request.

Table 95. Response from a *get tenant* request (continued)

Field	Type	Description
tenant-consumer-list	Array of Strings	Consumer user IDs for the tenant.
object-uri	String	URI of the newly created object.
tenant-description	String	Description of the tenant.
tenant-metering-capping-properties	JSON Object	<p>Properties of tenant capping:</p> <p>tenant-capping-enabled Indicates if capping for the tenant is enabled. Values are true or false.</p> <p>tenant-metering-enabled Indicates if metering for the tenant is enabled. Values are true or false.</p> <p>tenant-cpu-cap-type Indicates the type of capping for the tenant. Values are:</p> <p>lpar-share-percentage Percentage of the LPAR share in the general purpose processor pool.</p> <p>service-unit Unweighted CPU service units per second.</p> <p>cp A number of general purpose processors (CPs), including numbers with up to two decimal places.</p> <p>msu Millions of service units per hour.</p> <p>none Removes all of the capping properties.</p> <p>tenant-cpu-cap-limit Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type:</p> <p>lpar-share-percentage 1-100.</p> <p>service-unit 1-999999999.</p> <p>cp 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP.</p> <p>msu 1-999999999.</p> <p>tenant-solution-id Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.</p>
tenant-resource-group-name	String	Name of the tenant resource group, which can be used for processor capping or container pricing.

Table 95. Response from a get tenant request (continued)

Field	Type	Description
tenant-group-list	Array of Strings	List of groups in the tenant.
tenant-state	String	<p>State of the tenant.</p> <p>security_update_failed Indicates that the security workflow that provides automatic security failed. The accompanying error message indicates the workflow name and workflow key. To understand why the security workflow failed, use the z/OSMF Workflows task to review the failed workflow step status and the workflow history. Make corrections as necessary, then use the Set Security Complete action for the domain.</p> <p>Pending Security Update indicates one of the following:</p> <p>pending_security_update indicates one of the following:</p> <ul style="list-style-type: none"> • Manual Security definition was selected for the domain, and security setup is required. • Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. Make corrections as necessary, then use the Set Security Complete action for the tenant. <p>wlm-update-failed Indicates that an attempt to modify the Workload Management (WLM) service definition that is associated with the tenant failed. The attempted modification included one of these:</p> <ul style="list-style-type: none"> • Specifying a Solution ID, enabling metering, or enabling capping • Modifying existing Workload Management resource pools. <p>Review the accompanying error messages, make corrections as necessary and use the Set Security Complete action to try the Workload Management modification for the tenant and accompanying Workload Management resource pools again. Or, reverse the modification (for example, disable metering) and, if necessary, use the Set Security Complete action to return the state to Operational.</p> <p>operational Indicates that the tenant is ready for use.</p>
create-time	String	Date and time that the tenant was created.
created-by-user	String	User who created the tenant.
last-modified-time	String	The date and time of the last modification to the tenant.
last-modified-by-user	String	User who last modified the tenant.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 97 on page 125 .

Table 95. Response from a get tenant request (continued)

Field	Type	Description
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.

Table 96. Response from a get request: tenant-templates

Field	Type	Description
rdp-id	String	Identifier of the resource deployment pool.
template-available	boolean	Deprecated and does not have accurate information.
template-name	String	Name of the template.

Table 97. Response from a create request: SAF-resource object

Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Example HTTP interaction

In [Figure 38](#) on page 126, a request is submitted to retrieve a tenant.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/<object-id>
```

Figure 38. Sample request to get a tenant

The following is the response body for the example get tenant request.

```
{
  "tenant-id": "IYU100",
  "tenant-name": "Tenant",
  "tenant-shared-rdp-id": "IYU100ZZ",
  "tenant-domain-id": "IYU1",
  "tenant-domain-name": "Domain1",
  "tenant-state": "operational",
  "tenant-metering-capping-properties": {
    "tenant-metering-enabled": false,
    "tenant-capping-enabled": false,
    "tenant-cpu-cap-type": "",
    "tenant-cpu-cap-limit": 0.0
  },
  "tenant-templates": [{
    "template-name": "Template",
    "rdp-id": "IYU10000",
    "template-available": false
  }],
  "tenant-consumer-list": ["consumer"],
  "tenant-group-list": ["group"],
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100",
  "tenant-description": "Tenant description",
  "create-time": "2017-10-18T20:27:49.723Z",
  "created-by-user": "PROVADM",
  "last-modified-time": "2017-10-18T20:33:00.676Z",
  "last-modified-by-user": "PROVADM",
  "SAF-resources": [{
    "description": "Designates the user as a z/OSMF user with authorization to log in.",
    "ids": ["consumer"],
    "groups": ["group"],
    "role": "Tenant Consumer",
    "resource-class": "ZMFAPLA",
    "resource-name": "IZUDFLT.ZOSMF",
    "required-access": "SAF_READ",
    "other-required-ids": [],
    "audit-requirements": ""
  },
  ,
  {, ...}],
  "provisioning-version": "1400",
  "provisioning-version-supported": true
}
```

Get a tenant history

Use this operation to retrieve a tenant history.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/history
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<tenant-id>

Identifies the tenant for which history is to be retrieved.

Query parameters

None.

Description

This operation retrieves a tenant history.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 127](#).

Request content

None.

Authorization requirements

The user must be a domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 127](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 98. HTTP error response codes for a get tenant history request	
HTTP error status code	Description
HTTP 404 Not found	The requested tenant does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the tenant. [Table 99 on page 128](#) lists the fields in the history object.

Table 99. Response from a get request: History object

Field	Type	Description
action-type	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"> • Create • Add consumer • Add template • Assign CPU capping • Assign memory capping • Assign solution ID • Create pool • Delete pool • Disable CPU capping • Disable memory capping • Disable metering • Enable CPU capping • Enable memory capping • Enable metering • Remove consumer • Remove template • Update description • Update state
user	String	The user who performed the action.
action-time	String	The time that the action was taken.
action-details	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

Example HTTP interaction

In Figure 39 on page 128, a request is submitted to retrieve the history for the tenant IYU100.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/history
```

Figure 39. Sample request to get a tenant history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:33:49.010Z",
      "action-details": "Created tenant"
    },
    {
      "action-type": "Create pool",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:34:26.514Z",
```

```
    "action-details": "Created tenant resource pool: d1.t1.*"  
  }  
]  
}
```

List the tenants

Use this operation to list the tenants that are defined for IBM Cloud Provisioning and Management for z/OS.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation lists the tenants that are defined for IBM Cloud Provisioning and Management for z/OS.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 130](#).

Request content

None.

Authorization requirements

The user must be the domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 100. HTTP error response codes for a list tenants request	
HTTP error status code	Description
HTTP 404 Not found	The domain does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the tenants. See [Table 101 on page 130](#).

Table 101. Response from a list tenants request		
Field	Type	Description
tenant-list	Array	Information about the tenants that are defined. See Table 102 on page 130 .

Table 102. Tenant list		
Field	Type	Description
tenant-id	String	Generated ID for the tenant.
tenant-name	String	Descriptive name for the tenant.
tenant-shared-rdp-id	String	<p>The ID of the shared resource pool that is associated with this tenant. It is empty until a shared resource pool is created.</p> <p>The resource pool ID suffix ZZ is reserved for the shared resource pool of the tenant. Dedicated resource pools cannot have ZZ as an ID suffix.</p>
tenant-domain-id	String	Generated ID for the domain to which the tenant belongs.
tenant-domain-name	String	Descriptive name of the domain to which the tenant belongs.
tenant-templates	Array of Strings	Array that describes the templates that are associated with dedicated resource pools. See Table 96 on page 125 . For templates that are associated with shared resource pools, see the rdp-shared-template-name-list field in a Get resource pool request.
tenant-consumer-list	Array of Strings	Consumer user IDs for the tenant.
object-uri	String	URI of the newly created object.
tenant-description	String	Description of the tenant.

Table 102. Tenant list (continued)

Field	Type	Description
tenant-metering-capping-properties	JSON Object	<p>Properties of tenant capping:</p> <p>tenant-capping-enabled Indicates if capping for the tenant is enabled. Values are true or false.</p> <p>tenant-metering-enabled Indicates if metering for the tenant is enabled. Values are true or false.</p> <p>tenant-cpu-cap-type Indicates the type of capping for the tenant. Values are:</p> <p>lpar-share-percentage Percentage of the LPAR share in the general purpose processor pool.</p> <p>service-unit Unweighted CPU service units per second.</p> <p>cp A number of general purpose processors (CPs), including numbers with up to two decimal places.</p> <p>msu Millions of service units per hour.</p> <p>none Removes all of the capping properties.</p> <p>tenant-cpu-cap-limit Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type:</p> <p>lpar-share-percentage 1-100.</p> <p>service-unit 1-999999999.</p> <p>cp 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP.</p> <p>msu 1-999999999.</p> <p>tenant-solution-id Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.</p>
tenant-resource-group-name	String	Name of the tenant resource group, which can be used for processor capping or container pricing.

<i>Table 102. Tenant list (continued)</i>		
Field	Type	Description
tenant-group-list	Array of Strings	List of groups in the tenant.

Table 102. Tenant list (continued)

Field	Type	Description
tenant-state	String	<p>State of the tenant.</p> <p>security_update_failed Indicates that the security workflow that provides automatic security failed. The accompanying error message indicates the workflow name and workflow key. To understand why the security workflow failed, use the z/OSMF Workflows task to review the failed workflow step status and the workflow history. Make corrections as necessary, then use the Set Security Complete action for the domain.</p> <p>Pending Security Update indicates one of the following:</p> <p>pending_security_update indicates one of the following:</p> <ul style="list-style-type: none"> • Manual Security definition was selected for the domain, and security setup is required. • Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. Make corrections as necessary, then use the Set Security Complete action for the tenant. <p>wlm-update-failed Indicates that an attempt to modify the Workload Management (WLM) service definition that is associated with the tenant failed. The attempted modification included one of these:</p> <ul style="list-style-type: none"> • Specifying a Solution ID, enabling metering, or enabling capping • Modifying existing Workload Management resource pools. <p>Review the accompanying error messages, make corrections as necessary and use the Set Security Complete action to try the Workload Management modification for the tenant and accompanying Workload Management resource pools again. Or, reverse the modification (for example, disable metering) and, if necessary, use the Set Security Complete action to return the state to Operational.</p> <p>operational Indicates that the tenant is ready for use.</p>

Table 102. Tenant list (continued)		
Field	Type	Description
create-time	String	Date and time that the tenant was created.
created-by-user	String	User who created the tenant.
last-modified-time	String	The date and time of the last modification to the tenant.
last-modified-by-user	String	User who last modified the tenant.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 97 on page 125 .
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.

Example HTTP interaction

In [Figure 40 on page 134](#), a request is submitted to list the tenants.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/
```

Figure 40. Sample request to list tenants

The following is the response body for the example list tenants request.

```
{
  "tenant-list": [{
    "tenant-id": "IYU100",
    "tenant-name": "Tenant",
    "tenant-shared-rdp-id": "IYU100ZZ",
    "tenant-domain-id": "IYU1",
    "tenant-domain-name": "Domain1",
    "tenant-state": "operational",
    "tenant-metering-capping-properties": {
      "tenant-metering-enabled": false,
      "tenant-capping-enabled": false,
      "tenant-cpu-cap-type": "",
      "tenant-cpu-cap-limit": 0.0
    },
    "tenant-templates": [{
      "template-name": "Template",
      "rdp-id": "IYU10000",
      "template-available": false
    }],
    "tenant-consumer-list": ["consumer"],
    "tenant-group-list": ["group"],
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100",
    "tenant-description": "",
    "create-time": "2017-10-18T20:29:58.963Z",
    "created-by-user": "PROVADM",
    "last-modified-time": "2017-10-18T20:37:23.046Z",
    "last-modified-by-user": "PROVADM",
    "SAF-resources": [{
      "description": "Designates the user as a z/OSMF user with authorization to log in.",

```

```

        "ids": ["consumer"],
        "groups": ["group"],
        "role": "Tenant Consumer",
        "resource-class": "ZMFAPLA",
        "resource-name": "IZUDFLT.ZOSMF",
        "required-access": "SAF_READ",
        "other-required-ids": [],
        "audit-requirements": "",
    }, ...],
    "provisioning-version": "1400",
    "provisioning-version-supported": true
  }
}

```

Delete a tenant

Use this operation to delete the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation deletes the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the deletion of the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 103. HTTP error response codes for a delete tenant request

HTTP error status code	Description
HTTP 204	The request was processed successfully. However, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 41 on page 136, a request is submitted to delete the tenant IYU100.

```
DELETE https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100
```

Figure 41. Sample request to delete a tenant

Assign CPU properties to a tenant

Use this operation to assign a CPU capping type and limit to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-cpu-capping-properties
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation assigns the specified tenant a CPU capping type and CPU capping limit.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in CPU properties being assigned to the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 104 on page 137](#).

Table 104. Request content for the assign CPU properties to a tenant request			
Field name	Type	Required or optional	Description
tenant-cpu-cap-limit	Integer	Required	Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type: lpar-share-percentage 1-100. service-unit 1-999999999. cp 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP. msu 1-999999999.
tenant-cpu-cap-type	String	Required	Indicates the type of capping for the tenant. Values are: lpar-share-percentage Percentage of the LPAR share in the general purpose processor pool. service-unit Unweighted CPU service units per second. cp A number of general purpose processors (CPs), including numbers with up to two decimal places. msu Millions of service units per hour. none Removes all of the capping properties.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 105. HTTP error response codes for an assign CPU properties to a tenant request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 42 on page 138, a request is submitted to assign CPU properties to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-cpu-capping-properties
{
  "tenant-cpu-cap-limit": 0,
  "tenant-cpu-cap-type": "lpar-share-percentage"
}
```

Figure 42. Sample request to assign CPU properties to a tenant, with the request body

Assign memory capping properties to a tenant

Use this operation to assign memory capping properties to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-memory-capping-properties
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation assigns memory capping properties to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in memory capping properties being assigned to the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 106 on page 139](#).

Table 106. Request content for the assign memory capping properties to a tenant request			
Field name	Type	Required or optional	Description
tenant-memory-cap-limit	Integer	Required	The limit in gigabytes of the memory cap.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 107. HTTP error response codes for an assign CPU properties to a tenant request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 43 on page 140, a request is submitted to assign a memory capping limit of 0 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-memory-capping-properties
{
  "tenant-memory-cap-limit": 0
}
```

Figure 43. Sample request to assign memory capping properties to a tenant, with the request body

Assign a solution ID

Use this operation to assign a solution ID to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-solution-id
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation assigns a solution ID to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the assignment of a solution ID to the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See Table 108 on page 140.

Table 108. Request content for the assign solution ID request

Field name	Type	Required or optional	Description
tenant-solution-id	String	Required	The solution ID that corresponds to your Container Pricing for IBM Z solution. This value is defined in the License Management Support (LMS) web portal and must be exactly 64 characters.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 109. HTTP error response codes for an assign solution ID request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 44 on page 141, a request is submitted to assign a solution ID "string" to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-solution-id
{
  "tenant-solution-id": "string"
}
```

Figure 44. Sample request to assign a solution ID, with the request body

Disable CPU capping

Use this operation to disable CPU capping for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-cpu-capping
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation disables CPU capping for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of CPU capping for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102.](#)

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 110. HTTP error response codes for a disable CPU capping request</i>	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 45 on page 143, a request is submitted to disable CPU capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-cpu-capping
```

Figure 45. Sample request to disable CPU capping

Disable memory capping

Use this operation to disable memory capping for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-memory-capping
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation disables memory capping for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of memory capping for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 111. HTTP error response codes for a disable memory capping request

HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 46 on page 144, a request is submitted to disable memory capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-memory-capping
```

Figure 46. Sample request to disable memory capping

Disable metering

Use this operation to disable metering for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-metering
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation disables metering for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of metering for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 112. HTTP error response codes for a disable metering request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 47 on page 145](#), a request is submitted to disable metering for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-metering
```

Figure 47. Sample request to disable metering

Enable CPU capping

Use this operation to enable CPU capping for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-cpu-capping
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation enables CPU capping for the specified tenant.

Important: To enable CPU capping for a tenant, you must first perform the 'assign-cpu-capping-properties' action to assign a CPU capping type and CPU capping limit to the tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of CPU capping for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 113. HTTP error response codes for an enable CPU capping request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.

Table 113. HTTP error response codes for an enable CPU capping request (continued)	
HTTP error status code	Description
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 48 on page 147, a request is submitted to enable CPU capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-cpu-capping
```

Figure 48. Sample request to enable CPU capping

Enable memory capping

Use this operation to enable memory capping for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-memory-capping
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation enables memory capping for the specified tenant.

Important: To enable memory capping for a tenant, you must first perform the 'assign-memory-capping-properties' action to assign a memory capping limit to the tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of memory capping for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 114. HTTP error response codes for an enable memory capping request</i>	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 49 on page 148](#), a request is submitted to enable memory capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-memory-capping
```

Figure 49. Sample request to enable memory capping

Enable metering

Use this operation to enable metering for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-metering
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation enables metering for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of metering for the specified tenant.

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 115. HTTP error response codes for an enable metering request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.

Table 115. HTTP error response codes for an enable metering request (continued)

HTTP error status code	Description
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 50 on page 150, a request is submitted to enable metering for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-metering
```

Figure 50. Sample request to enable metering

Add tenant consumer

Use this operation to add a list of one or more consumers to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/  
actions/add
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation adds a list of one or more consumers to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the addition of the specified consumers to the specified tenant.

Request content

The request content is expected to contain a JSON object that lists the consumer user IDs to be added to the tenant. See [Table 116 on page 151](#).

Table 116. Request content for the add tenant consumer request

Field name	Type	Required or optional	Description
tenant-consumer-list	String	Required	Consumer user IDs to add to the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 117. HTTP error response codes for an add tenant consumer request

HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 51 on page 151](#), a request is submitted to add the user IDs consumer1 and consumer2 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/consumers/actions/add
{
  "tenant-consumer-list": ["consumer1", "consumer2"]
}
```

Figure 51. Sample request to add consumers to a tenant, with the request body

Remove tenant consumer

Use this operation to remove a list of one or more consumers from the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/remove
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation removes a list of one or more consumers from the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the removal of the one or more listed tenant consumers from the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 118 on page 152](#).

Table 118. Request content for the remove tenant consumer request			
Field name	Type	Required or optional	Description
tenant-consumer-list	String	Required	Consumer user IDs to remove from the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 119. HTTP error response codes for a remove tenant consumer request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.

Table 119. HTTP error response codes for a remove tenant consumer request (continued)	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 52 on page 153, a request is submitted to remove `consumer1` and `consumer2` from the tenant `IYU100`.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/consumers/actions/remove
{
  "tenant-consumer-list": ["consumer1", "consumer2"]
}
```

Figure 52. Sample request to remove tenant consumer, with the request body

Add tenant description

Use this operation to add a description to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/description/actions/add
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation adds a description to the specified tenant. If the tenant already has a description, this operation overwrites the existing description.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in addition of a tenant description to the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 120 on page 154](#).

Table 120. Request content for the add tenant description request			
Field name	Type	Required or optional	Description
tenant-description	String	Required	The description to add to the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 121. HTTP error response codes for an add tenant description request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 53 on page 155](#), a request is submitted to add the description "This group of users has the authority to provision software instances." to the tenant IYU100.


```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/description/actions/add
{
  "tenant-description": "This group of users has the authority to provision software instances."
}
```

Figure 53. Sample request to add tenant description, with the request body

Add tenant groups

Use this operation to add tenant groups to the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/add
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation adds tenant groups to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the addition of the tenant groups to the specified tenant.

Request content

The request content is expected to contain a JSON object that is a list of groups to add to the tenant. See [Table 122 on page 155](#).

Table 122. Request content for the add tenant groups request			
Field name	Type	Required or optional	Description
tenant-group-list	String	Required	List of groups to add to the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 123. HTTP error response codes for an add tenant groups request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In Figure 54 on page 156, a request is submitted to add group1 and group2 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/groups/actions/add
{
  "tenant-group-list": [ "group1", "group2" ]
}
```

Figure 54. Sample request to add tenant groups, with the request body

Remove tenant groups

Use this operation to remove one or more groups from the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/remove
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation removes one or more groups from the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the removal of the tenant groups from the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 124 on page 157](#).

Table 124. Request content for the remove tenant groups request			
Field name	Type	Required or optional	Description
tenant-group-list	String	Required	The list of groups to remove from the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 125. HTTP error response codes for a remove tenant groups request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 55 on page 158](#), a request is submitted to remove group1 and group2 from the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/groups/actions/remove
{
  "tenant-group-list": [ "group1", "group2" ]
}
```

Figure 55. Sample request to remove tenant groups, with the request body

Get a resource pool

Use this operation to retrieve a resource pool.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/<rdp-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<tenant-id>

Identifies the tenant that the resource pool is associated with.

<rdp-id>

Identifies the resource pool to be retrieved.

Query parameters

None.

Description

This operation retrieves a resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 159](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 126. HTTP error response codes for a get resource pool request

HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 127 on page 159](#).

Table 127. Response from a get resource pool request

Field	Type	Description
rdp-id	String	The generated ID for the resource pool.
rdp-name	String	Descriptive name for the resource pool, in the form <i>domain-name.tenant-name</i> . For shared resource pools, the name ends with an asterisk (*).
rdp-pool-type	String	Type of resource pool: rdp-dedicated Dedicated to a single software services template rdp-shared Shared amongst software services templates
rdp-shared-service-instance-snaapplid-name-prefix	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-subsystem-name-prefix	String	The software services instance name subsystem prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-general-name-prefix	String	The software services instance name general prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-domain-id	String	The generated ID for the domain to which the resource pool belongs.
rdp-tenant-id	String	The generated ID for the tenant to which the resource pool belongs.
rdp-template-name	String	Name of the software services template that the dedicated resource pool is associated with.
rdp-shared-template-name-list	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.

Table 127. Response from a get resource pool request (continued)

Field	Type	Description
rdp-instance-limit	Integer	Maximum number of software services instances that are allowed for the template.
rdp-instance-actual	Integer	Actual number of software services instances for the software services template that exist.
rdp-system-instance	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
rdp-user-instance-limit	Integer	Maximum number of software services instances that are allowed for a single user.
rdp-instance-expiration-limit	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
rdp-registry-instance-list	Array of Strings	Array of strings, where each string is the identifier of a registry instance that is associated with the resource pool.
rdp-system-pool	Array of JSON objects	Array that describes the systems selected for provisioning. See Table 128 on page 163 .
rdp-system-selection	String	Type of system selection for the resource pool: single Use a specific system. auto Assign a system automatically. prompt Prompt the user for the system.
rdp-ready	boolean	Indicates if the resource pool is ready for use. true The resource pool is ready for use. false The resource pool is not ready for use.
rdp-network-resources-needed	boolean	Indicates if a network resource pool is required. true A network resource pool is required. false A network resource pool is not required.
rdp-network-pool-id	String	Generated identifier of the network resource pool.
rdp-network-pool-ready	String	Indicates if the network resource pool is ready for use. complete The network resource pool is ready for use. incomplete The network resource pool is not yet ready for use. attention The network resource pool requires attention.

Table 127. Response from a get resource pool request (continued)

Field	Type	Description
rdp-relocatable-resources	String	Optional, indicates if the resource pool can be relocated to a different system. movable The resource pool can be relocated to a different system. none Not specified.
rdp-sna-applid-prefixed-instance-name	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID. true The prefix should be derived from the SNA application ID. false The prefix should not be derived from the SNA application ID.
rdp-service-instance-name-prefix	String	The explicit prefix of the software services instance names.
rdp-wlm-resources-needed	boolean	Indicates if a workload management resource pool is required. true A workload management resource pool is required. false A workload management resource pool is not required.
rdp-wlm-sla	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
rdp-wlm-pool-id	String	Generated identifier of the workload management resource pool.
rdp-wlm-pool-ready	boolean	Indicates if the workload management resource pool is ready for use. complete The workload management resource pool is ready for use. incomplete The workload management resource pool is not yet ready for use.
rdp-report-class-name	String	The name of the workload management report class.
rdp-lpar-resources-needed	boolean	Indicates whether an LPAR resource pool is required. true An LPAR resource pool is required. false An LPAR resource pool is not required.

Table 127. Response from a get resource pool request (continued)

Field	Type	Description
rdp-lpar-pool-ready	boolean	Indicates whether the LPAR resource pool is ready for use. complete LPAR resource pool is ready for use. incomplete LPAR resource pool is not yet ready for use.
rdp-job-statement	String	JOB statement JCL that is used in provisioning jobs.
rdp-share-tenant-instances	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool. true Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool. false Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.
rdp-account-modify	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. true The account information can be modified. false The account information cannot be modified .
create-time	String	Date and time that the resource pool was created.
created-by-user	String	User ID of the user that created the resource pool.
last-modified-time	String	Date and time that the resource pool was most recently modified.
last-modified-by-user	String	User ID of the user who last modified the resource pool.
object-uri	String	URI of the newly resource pool object.
local-system	Array	Array that describes the local system. See Table 128 on page 163 .
rdp-quiesced	boolean	Indicates if the resource pool is quiesced: true The resource pool is quiesced. You cannot provision any resources for this pool. false The resource pool is not quiesced. You can provision any resources for this pool.
rdp-tenant-report-class-name	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.

Table 127. Response from a get resource pool request (continued)

Field	Type	Description
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.
rdp-composite-cluster	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: <p>true</p> The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools.
		<p>false</p> The resource pool is not to be used in a composite cluster.
rdp-cluster-name-prefix	String	The prefix used for cluster names.

Table 128. Response from a get request: Systems

Field	Type	Description
sysplex-name	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
sysplex-node-name	String	Sysplex node name.
system-nickname	String	Unique name that is assigned to the system definition.

Example HTTP interaction

In Figure 56 on page 163, a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/IYU100000
```

Figure 56. Sample request to get a resource pool

The following is the response body for the example get resource pool request.

```
Updated Response Body:
{
  "rdp-id": "IYU100000",
  "rdp-name": "Domain1.Tenant.Template",
  "rdp-pool-type": "rdp-dedicated",
  "rdp-quiesced": false,
  "rdp-domain-id": "IYU1",
  "rdp-tenant-id": "IYU100",
  "rdp-template-name": "Template",
  "rdp-instance-limit": 300,
  "rdp-instance-actual": 0,
  "rdp-system-instance": {
```

```

    "SY1": 0
  },
  "rdp-user-instance-limit": 12,
  "rdp-instance-expiration-limit": 0,
  "rdp-system-pool": [
    {
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SY1",
      "system-nickname": "SY1"
    }
  ],
  "rdp-system-selection": "single",
  "rdp-ready": true,
  "rdp-network-resources-needed": false,
  "rdp-network-pool-id": "",
  "rdp-network-pool-ready": "incomplete",
  "rdp-sna-applid-prefixed-instance-name": false,
  "rdp-service-instance-name-prefix": "TEMPL",
  "rdp-wlm-resources-needed": false,
  "rdp-wlm-sla": [],
  "rdp-wlm-pool-id": "",
  "rdp-wlm-pool-ready": "incomplete",
  "rdp-job-statement": "",
  "rdp-account-modify": true,
  "rdp-share-tenant-instances": false,
  "create-time": "2017-10-19T20:37:23.044Z",
  "created-by-user": "PROVADM",
  "last-modified-time": "2017-12-19T20:37:23.055Z",
  "last-modified-by-user": "PROVADM",
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/IYU100000",
  "local-system": {
    "sysplex-name": "PLEX1",
    "sysplex-node-name": "SY1",
    "system-nickname": "SY1"
  },
  "rdp-composite-cluster": true,
  "rdp-cluster-name-prefix": "Y",
  "provisioning-version": "1200",
  "provisioning-version-supported": true
}

```

Get a domain resource pool

Use this operation to retrieve a domain-shared resource pool.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/<rdp-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<domain-id>

Identifies the domain that the resource pool is associated with.

<rdp-id>

Identifies the resource pool to be retrieved.

Query parameters

None.

Description

This operation retrieves a domain-shared resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 165](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator for the domain that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 129. HTTP error response codes for a get domain resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 130 on page 165](#).

Table 130. Response from a get domain resource pool request		
Field	Type	Description
rdp-id	String	The generated ID for the resource pool.
rdp-name	String	Description name of the domain-shared resource pool. The resource pool name is in the form <i>domain-name.**</i> , where asterisks (**) are used to indicate that the resource pool is shared across the domain.
rdp-pool-type	String	Type of resource pool: rdp-dedicated Dedicated to a single software services template. rdp-shared Shared among software services templates.
rdp-shared-service-instance-snaapplid-name-prefix	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool. This property is returned only for shared resource pools.

Table 130. Response from a get domain resource pool request (continued)

Field	Type	Description
rpd-shared-service-instance-subsystem-name-prefix	String	The software services instance name subsystem prefix for a shared resource pool. This property is returned only for shared resource pools.
rpd-shared-service-instance-general-name-prefix	String	The software services instance name general prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-domain-id	String	The generated ID for the domain to which the resource pool belongs.
rdp-tenant-id	String	The generated ID for the tenant to which the resource pool belongs.
rdp-template-name	String	Name of the software services template that the dedicated resource pool is associated with.
rdp-shared-template-name-list	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
rdp-instance-limit	Integer	Maximum number of software services instances that are allowed for the template.
rdp-instance-actual	Integer	Actual number of software services instances for the software services template that exist.
rdp-system-instance	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
rdp-user-instance-limit	Integer	Maximum number of software services instances that are allowed for a single user.
rdp-instance-expiration-limit	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
rdp-registry-instance-list	Array of Strings	Array of strings, where each string is the identifier of a registry instance that is associated with the resource pool.
rdp-system-pool	Array of JSON objects	Array that describes the systems that are selected for provisioning. See Table 131 on page 169 .
rdp-system-selection	String	Type of system selection for the resource pool: single Use a specific system. auto Assign a system automatically. prompt Prompt the user for the system.

Table 130. Response from a get domain resource pool request (continued)

Field	Type	Description
rdp-ready	Boolean	Indicates whether the resource pool is ready for use. true The resource pool is ready for use. false The resource pool is not ready for use.
rdp-network-resources-needed	Boolean	Indicates whether a network resource pool is required. true A network resource pool is required. false A network resource pool is not required.
rdp-network-pool-id	String	Generated identifier of the network resource pool.
rdp-network-pool-ready	String	Indicates whether the network resource pool is ready for use. complete The network resource pool is ready for use. incomplete The network resource pool is not yet ready for use. attention The network resource pool requires attention.
rdp-relocatable-resources	String	Optional, indicates whether the resource pool can be relocated to a different system. movable The resource pool can be relocated to a different system. none Not specified.
rdp-sna-applid-prefixed-instance-name	Boolean	Indicates whether the prefix of the software services instance names should be derived from the SNA application ID. true The prefix should be derived from the SNA application ID. false The prefix should not be derived from the SNA application ID.
rdp-service-instance-name-prefix	String	The explicit prefix of the software services instance names.
rdp-wlm-resources-needed	Boolean	Indicates whether a workload management resource pool is required. true A workload management resource pool is required. false A workload management resource pool is not required.

Table 130. Response from a get domain resource pool request (continued)

Field	Type	Description
rdp-wlm-sla	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
rdp-wlm-pool-id	String	Generated identifier of the workload management resource pool.
rdp-wlm-pool-ready	Boolean	Indicates whether the workload management resource pool is ready for use. complete The workload management resource pool is ready for use. incomplete The workload management resource pool is not yet ready for use.
rdp-report-class-name	String	The name of the workload management report class.
rdp-lpar-resources-needed	boolean	Indicates whether an LPAR resource pool is required. true An LPAR resource pool is required. false An LPAR resource pool is not required.
rdp-lpar-pool-ready	boolean	Indicates whether the LPAR resource pool is ready for use. complete LPAR resource pool is ready for use. incomplete LPAR resource pool is not yet ready for use.
rdp-job-statement	String	JOB statement JCL that is used in provisioning jobs.
rdp-share-tenant-instances	Boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool. true Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool. false Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.
rdp-account-modify	Boolean	Indicates whether the account information can be modified when a template is provisioned, with a Test Run or Run action. true The account information can be modified. false The account information cannot be modified.

Table 130. Response from a *get* domain resource pool request (continued)

Field	Type	Description
create-time	String	Date and time that the resource pool was created.
created-by-user	String	User ID of the user that created the resource pool.
last-modified-time	String	Date and time that the resource pool was most recently modified.
last-modified-by-user	String	User ID of the user who last modified the resource pool.
object-uri	String	URI of the newly created resource pool object.
local-system	Array	Array that describes the local system. See Table 128 on page 163 .
rdp-quieted	Boolean	Indicates whether the resource pool is quiesced: true The resource pool is quiesced. You cannot provision any resources for this pool. false The resource pool is not quiesced. You can provision any resources for this pool.
rdp-tenant-report-class-name	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.
rdp-composite-cluster	Boolean	Indicates whether the resource pool is to be used in a composite cluster, as follows: true The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. false The resource pool is not to be used in a composite cluster.
rdp-cluster-name-prefix	String	The prefix used for cluster names.

Table 131. Response from a *get* request: Systems

Field	Type	Description
sysplex-name	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
sysplex-node-name	String	Sysplex node name.

Table 131. Response from a get request: Systems (continued)

Field	Type	Description
system-nickname	String	Unique name that is assigned to the system definition.

Example HTTP interaction

In “Get a domain resource pool” on page 164, a request is submitted to retrieve the resource pool IYU2ZZZZ from domain domain2.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ
```

Figure 57. Sample request to get a domain resource pool

The following figure shows the response body for the get domain resource pool request in the previous example.

```
Updated Response Body: {
  "rdp-id": "IYU1ZZZZ",
  "rdp-name": "d1.*.*",
  "rdp-pool-type": "rdp-shared",
  "rdp-quiesced": false,
  "rdp-domain-id": "IYU1",
  "rdp-tenant-id": "IYU1ZZ",
  "rdp-template-name": "*",
  "rdp-instance-limit": 123,
  "rdp-instance-actual": 0,
  "rdp-system-instance": {
    "SY1": 0
  },
  "rdp-user-instance-limit": 123,
  "rdp-system-pool": [
    {
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SY1",
      "system-nickname": "SY1"
    }
  ],
  "rdp-system-selection": "single",
  "rdp-ready": true,
  "rdp-network-resources-needed": false,
  "rdp-network-pool-id": "",
  "rdp-network-pool-ready": "incomplete",
  "rdp-network-pool-local-ready": "incomplete",
  "rdp-sna-applid-prefixed-instance-name": false,
  "rdp-service-instance-name-prefix": "",
  "rdp-wlm-resources-needed": false,
  "rdp-wlm-sla": [],
  "rdp-wlm-pool-id": "",
  "rdp-wlm-pool-ready": "incomplete",
  "rdp-wlm-pool-local-ready": "incomplete",
  "rdp-job-statement": "",
  "rdp-account-modify": true,
  "rdp-shared-service-instance-snaapplid-name-prefix": "",
  "rdp-shared-service-instance-subsystem-name-prefix": "SP",
  "rdp-shared-service-instance-general-name-prefix": "SG",
  "rdp-share-tenant-instances": false,
  "rdp-composite-cluster": false,
  "rdp-cluster-name-prefix": null,
  "rdp-storage-resources-needed": false,
  "rdp-storage-pool": {
    "dataset-attributes-list": []
  },
  "rdp-instance-expiration-limit": 23,
  "create-time": "2020-12-14T16:11:27.135Z",
  "created-by-user": "ibmuser",
  "last-modified-time": "2020-12-14T16:11:27.167Z",
  "last-modified-by-user": "ibmuser",
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ",
  "local-system": {

```



```
        "sysplex-name": "PLEX1",
        "sysplex-node-name": "SY1",
        "system-nickname": "SY1"
      },
      "provisioning-version": "1600",
      "provisioning-version-supported": true
    }
  }
```

Get a resource pool history

Use this operation to retrieve a resource pool history.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/history
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<rdp-id>

Identifies the resource pool for which history is to be retrieved.

Query parameters

None.

Description

This operation retrieves the history for a resource pool.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 172](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 172](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 132. HTTP error response codes for a get resource pool request

HTTP error status code	Description
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the resource pool. [Table 133 on page 172](#) lists the fields in the history object.

Table 133. Response from a get request: History object

Field	Type	Description
action-type	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"> • Create • Add template • Modify • Quiesce • Remove template • Unquiesce • Add WLM classification rule • Create LPAR pool entry • Delete LPAR pool entry • Modify LPAR pool entry • Obtain IP • Obtain LPAR pool entry • Obtain port • Obtain SNA application name • Release IP • Release LPAR pool entry • Release port • Release SNA application name • Remove WLM classification rule
user	String	The user who performed the action.
action-time	String	The time that the action was taken.
action-details	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

Example HTTP interaction

In [Figure 58 on page 173](#), a request is submitted to retrieve the history for the resource pool IYU10000.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/IYU10000/history
```

Figure 58. Sample request to get a resource pool history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:34:26.502Z",
      "action-details": "Created resource pool"
    },
    {
      "action-type": "Add template",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:39.285Z",
      "action-details": "Added template template1, tenant: t1"
    }
  ]
}
```

List the resource pools

Use this operation to list the resource pools.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<tenant-id>

Identifies the tenant that the resource pool is associated with.

<rdp-id>

Identifies the resource pool to be retrieved.

Query parameters

None.

Description

This operation lists the resource pools for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 174](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 134. HTTP error response codes for a get resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [“Response content” on page 174](#).

Table 135. Response from a list resource pool request		
Field	Type	Description
rdp-list	Array	Array describing the resource pools. See Table 136 on page 174 .

Table 136. Resource pool list		
Field	Type	Description
rdp-id	String	The generated ID for the resource pool.
rdp-name	String	Descriptive name for the resource pool, in the form <i>domain-name.tenant-name</i> . For shared resource pools, the name ends with an asterisk (*).
rdp-pool-type	String	Type of resource pool: rdp-dedicated Dedicated to a single software services template rdp-shared Shared amongst software services templates
rdp-shared-service-instance-snaapplid-name-prefix	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool. This property is returned only for shared resource pools.

Table 136. Resource pool list (continued)

Field	Type	Description
rpd-shared-service-instance-subsystem-name-prefix	String	The software services instance name subsystem prefix for a shared resource pool. This property is returned only for shared resource pools.
rpd-shared-service-instance-general-name-prefix	String	The software services instance name general prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-domain-id	String	The generated ID for the domain to which the resource pool belongs.
rdp-tenant-id	String	The generated ID for the tenant to which the resource pool belongs.
rdp-template-name	String	Name of the software services template that the dedicated resource pool is associated with.
rdp-shared-template-name-list	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
rdp-instance-limit	Integer	Maximum number of software services instances that are allowed for the template.
rdp-instance-actual	Integer	Actual number of software services instances for the software services template that exist.
rdp-system-instance	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
rdp-user-instance-limit	Integer	Maximum number of software services instances that are allowed for a single user.
rdp-registry-instance-list	Array of Strings	Array of strings, where each string is the identifier of a registry instance that is associated with the resource pool.
rdp-system-pool	Array of JSON objects	Array that describes the systems selected for provisioning. See Table 128 on page 163 .
rdp-system-selection	String	Type of system selection for the resource pool: single Use a specific system. auto Assign a system automatically. prompt Prompt the user for the system.
rdp-ready	boolean	Indicates if the resource pool is ready for use. true The resource pool is ready for use. false The resource pool is not ready for use.

Table 136. Resource pool list (continued)

Field	Type	Description
rdp-network-resources-needed	boolean	Indicates if a network resource pool is required. true A network resource pool is required. false A network resource pool is not required.
rdp-network-pool-id	String	Generated identifier of the network resource pool.
rdp-network-pool-ready	String	Indicates if the network resource pool is ready for use. complete The network resource pool is ready for use. incomplete The network resource pool is not yet ready for use. attention The network resource pool requires attention.
rdp-relocatable-resources	String	Optional, indicates if the resource pool can be relocated to a different system. movable The resource pool can be relocated to a different system. none Not specified.
rdp-sna-applid-prefixed-instance-name	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID. true The prefix should be derived from the SNA application ID. false The prefix should not be derived from the SNA application ID.
rdp-service-instance-name-prefix	String	The explicit prefix of the software services instance names.
rdp-wlm-resources-needed	boolean	Indicates if a workload management resource pool is required. true A workload management resource pool is required. false A workload management resource pool is not required.
rdp-wlm-sla	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
rdp-wlm-pool-id	String	Generated identifier of the workload management resource pool.

Table 136. Resource pool list (continued)

Field	Type	Description
rdp-wlm-pool-ready	boolean	Indicates if the workload management resource pool is ready for use. complete The workload management resource pool is ready for use. incomplete The workload management resource pool is not yet ready for use.
rdp-report-class-name	String	The name of the workload management report class.
rdp-lpar-resources-needed	boolean	Indicates whether an LPAR resource pool is required. true An LPAR resource pool is required. false An LPAR resource pool is not required.
rdp-lpar-pool-ready	boolean	Indicates whether the LPAR resource pool is ready for use. complete LPAR resource pool is ready for use. incomplete LPAR resource pool is not yet ready for use.
rdp-job-statement	String	JOB statement JCL that is used in provisioning jobs.
rdp-share-tenant-instances	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool. true Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool. false Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.
rdp-account-modify	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. true The account information can be modified. false The account information cannot be modified .
create-time	String	Date and time that the resource pool was created.
created-by-user	String	User ID of the user that created the resource pool.
last-modified-time	String	Date and time that the resource pool was most recently modified.

Table 136. Resource pool list (continued)

Field	Type	Description
last-modified-by-user	String	User ID of the user who last modified the resource pool.
object-uri	String	URI of the newly resource pool object.
local-system	Array	Array that describes the local system. See Table 128 on page 163 .
rdp-quiesced	boolean	Indicates if the resource pool is quiesced: true The resource pool is quiesced. You cannot provision any resources for this pool. false The resource pool is not quiesced. You can provision any resources for this pool.
rdp-tenant-report-class-name	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.
rdp-composite-cluster	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: true The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. false The resource pool is not to be used in a composite cluster.
rdp-cluster-name-prefix	String	The prefix used for cluster names.

Example HTTP interaction

In [Figure 59 on page 178](#), a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/
```

Figure 59. Sample request to list resource pools

The following is the response body for the example get resource pool request.

```
{
  "rdp-list": [{
    "rdp-id": "IYU10100",
    "rdp-name": "domain1.tenant2.template1",
    "rdp-pool-type": "rdp-dedicated",
```



```

    "rdp-domain-id": "IYU1",
    "rdp-tenant-id": "IYU101",
    "rdp-template-name": "template1",
    "rdp-instance-limit": 101,
    "rdp-instance-actual": 0,
    "rdp-system-instance": {
      "DUMBNODE": 0
    },
    "rdp-user-instance-limit": 99,
    "rdp-system-pool": [{
      "sysplex-name": "DUMBPLEX",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE"
    }],
    "rdp-system-selection": "single",
    "rdp-ready": true,
    "rdp-network-resources-needed": false,
    "rdp-network-pool-id": "",
    "rdp-network-pool-ready": "incomplete",
    "rdp-relocatable-resources": "MOVABLE",
    "rdp-sna-applid-prefixed-instance-name": false,
    "rdp-service-instance-name-prefix": "TEMP",
    "rdp-wlm-resources-needed": false,
    "rdp-wlm-sla": [],
    "rdp-wlm-pool-id": "",
    "rdp-wlm-pool-ready": "incomplete",
    "rdp-report-class-name": "",
    "rdp-job-statement": "",
    "rdp-account-modify": true,
    "rdp-share-tenant-instances": false,
    "create-time": "2017-06-19T01:58:35.919Z",
    "created-by-user": "PROVADM",
    "last-modified-time": "2017-06-19T02:01:34.928Z",
    "last-modified-by-user": "PROVADM",
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU101/rdp/IYU10100",
    "local-system": {
      "sysplex-name": "DUMBPLEX",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE"
    }
  },
  "rdp-composite-cluster": true,
  "rdp-cluster-name-prefix": "Y",
  "provisioning-version": "1400",
  "provisioning-version-supported": true
}
}]
}

```

List domain resource pools

Use this operation to list the domain-shared resource pools.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<domain-id>

Identifies the domain that the resource pool is associated with.

<rdp-id>

Identifies the resource pool to be retrieved.

Query parameters

None.

Description

This operation lists the domain-shared resource pools for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 180](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator or a domain administrator for the domain that the resource pool is in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 137. HTTP error response codes for a get domain resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 138 on page 180](#).

Table 138. Response from a list domain resource pool request		
Field	Type	Description
rdp-list	Array	Array describing the resource pools. See “List domain resource pools” on page 179 .

Table 139. Domain resource pool list		
Field	Type	Description
rdp-id	String	The generated ID for the resource pool.
rdp-name	String	Description name of the domain-shared resource pool. The resource pool name is in the form <i>domain-name.**</i> , where asterisks (**) are used to indicate that the resource pool is shared across the domain.

Table 139. Domain resource pool list (continued)

Field	Type	Description
rdp-pool-type	String	Type of resource pool: rdp-dedicated Dedicated to a single software services template rdp-shared Shared amongst software services templates
rdp-shared-service-instance-snaapplid-name-prefix	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-subsystem-name-prefix	String	The software services instance name subsystem prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-general-name-prefix	String	The software services instance name general prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-domain-id	String	The generated ID for the domain to which the resource pool belongs.
rdp-tenant-id	String	The generated ID for the tenant to which the resource pool belongs.
rdp-template-name	String	Name of the software services template that the dedicated resource pool is associated with.
rdp-shared-template-name-list	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
rdp-instance-limit	Integer	Maximum number of software services instances that are allowed for the template.
rdp-instance-actual	Integer	Actual number of software services instances for the software services template that exist.
rdp-system-instance	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
rdp-user-instance-limit	Integer	Maximum number of software services instances that are allowed for a single user.
rdp-instance-expiration-limit	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
rdp-system-pool	Array of JSON objects	Array that describes the systems selected for provisioning. See Table 128 on page 163 .

Table 139. Domain resource pool list (continued)

Field	Type	Description
rdp-system-selection	String	Type of system selection for the resource pool: single Use a specific system. auto Assign a system automatically. prompt Prompt the user for the system.
rdp-ready	boolean	Indicates if the resource pool is ready for use. true The resource pool is ready for use. false The resource pool is not ready for use.
rdp-network-resources-needed	boolean	Indicates if a network resource pool is required. true A network resource pool is required. false A network resource pool is not required.
rdp-network-pool-id	String	Generated identifier of the network resource pool.
rdp-network-pool-ready	String	Indicates if the network resource pool is ready for use. complete The network resource pool is ready for use. incomplete The network resource pool is not yet ready for use. attention The network resource pool requires attention.
rdp-relocatable-resources	String	Optional, indicates if the resource pool can be relocated to a different system. movable The resource pool can be relocated to a different system. none Not specified.
rdp-sna-applid-prefixed-instance-name	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID. true The prefix should be derived from the SNA application ID. false The prefix should not be derived from the SNA application ID.
rdp-service-instance-name-prefix	String	The explicit prefix of the software services instance names.

Table 139. Domain resource pool list (continued)

Field	Type	Description
rdp-wlm-resources-needed	boolean	Indicates if a workload management resource pool is required. true A workload management resource pool is required. false A workload management resource pool is not required.
rdp-wlm-sla	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
rdp-wlm-pool-id	String	Generated identifier of the workload management resource pool.
rdp-wlm-pool-ready	boolean	Indicates if the workload management resource pool is ready for use. complete The workload management resource pool is ready for use. incomplete The workload management resource pool is not yet ready for use.
rdp-report-class-name	String	The name of the workload management report class.
rdp-lpar-resources-needed	boolean	Indicates whether an LPAR resource pool is required. true An LPAR resource pool is required. false An LPAR resource pool is not required.
rdp-lpar-pool-ready	boolean	Indicates whether the LPAR resource pool is ready for use. complete LPAR resource pool is ready for use. incomplete LPAR resource pool is not yet ready for use.
rdp-job-statement	String	JOB statement JCL that is used in provisioning jobs.
rdp-share-tenant-instances	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool. true Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool. false Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.

Table 139. Domain resource pool list (continued)

Field	Type	Description
rdp-account-modify	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. true The account information can be modified. false The account information cannot be modified .
create-time	String	Date and time that the resource pool was created.
created-by-user	String	User ID of the user that created the resource pool.
last-modified-time	String	Date and time that the resource pool was most recently modified.
last-modified-by-user	String	User ID of the user who last modified the resource pool.
object-uri	String	URI of the newly resource pool object.
local-system	Array	Array that describes the local system. See Table 128 on page 163 .
rdp-quiesced	boolean	Indicates if the resource pool is quiesced: true The resource pool is quiesced. You cannot provision any resources for this pool. false The resource pool is not quiesced. You can provision any resources for this pool.
rdp-tenant-report-class-name	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.
rdp-composite-cluster	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: true The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. false The resource pool is not to be used in a composite cluster.
rdp-cluster-name-prefix	String	The prefix used for cluster names.

Example HTTP interaction

In “List domain resource pools” on page 179, a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp
```

Figure 60. Sample request to list domain resource pools

The following is the response body for the request to list domain resource pools in the previous example.

```
{
  "rdp-list": [
    {
      "rdp-id": "IYU1ZZZZ",
      "rdp-name": "d1.*.*",
      "rdp-pool-type": "rdp-shared",
      "rdp-quiesced": false,
      "rdp-domain-id": "IYU1",
      "rdp-tenant-id": "IYU1ZZ",
      "rdp-template-name": "*",
      "rdp-instance-limit": 123,
      "rdp-instance-actual": 0,
      "rdp-system-instance": {
        "SY1": 0
      },
      "rdp-user-instance-limit": 123,
      "rdp-system-pool": [
        {
          "sysplex-name": "PLEX1",
          "sysplex-node-name": "SY1",
          "system-nickname": "SY1"
        }
      ],
      "rdp-system-selection": "single",
      "rdp-ready": true,
      "rdp-network-resources-needed": false,
      "rdp-network-pool-id": "",
      "rdp-network-pool-ready": "incomplete",
      "rdp-network-pool-local-ready": "incomplete",
      "rdp-sna-applid-prefixed-instance-name": false,
      "rdp-service-instance-name-prefix": "",
      "rdp-wlm-resources-needed": false,
      "rdp-wlm-sla": [],
      "rdp-wlm-pool-id": "",
      "rdp-wlm-pool-ready": "incomplete",
      "rdp-wlm-pool-local-ready": "incomplete",
      "rdp-job-statement": "",
      "rdp-account-modify": true,
      "rdp-shared-service-instance-snaapplid-name-prefix": "",
      "rdp-shared-service-instance-subsystem-name-prefix": "SP",
      "rdp-shared-service-instance-general-name-prefix": "SG",
      "rdp-share-tenant-instances": false,
      "rdp-composite-cluster": false,
      "rdp-cluster-name-prefix": null,
      "rdp-storage-resources-needed": false,
      "rdp-storage-pool": {
        "dataset-attributes-list": []
      },
      "rdp-instance-expiration-limit": 23,
      "create-time": "2020-12-14T16:11:27.135Z",
      "created-by-user": "ibmuser",
      "last-modified-time": "2020-12-14T16:11:27.167Z",
      "last-modified-by-user": "ibmuser",
      "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ",
      "local-system": {
        "sysplex-name": "PLEX1",
        "sysplex-node-name": "SY1",
        "system-nickname": "SY1"
      },
      "provisioning-version": "1600",
      "provisioning-version-supported": true
    }
  ]
}
```

List template resource pools

Use this operation to list the resource pool entries that are associated with a template in a domain.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/template/<template-name>/rdp
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

<domain-id>

Identifies the domain that the resource pool is associated with.

<template-name>

Identifies the template that the resource pool is associated with.

<rdp-id>

Identifies the resource pool to be retrieved.

Query parameters

None.

Description

This operation lists the domain-shared resource pools for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 187](#).

Request content

None.

Authorization requirements

The user must be a provisioning administrator, domain administrator, or a consumer for the tenants that the resource pools are in.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 140. HTTP error response codes for a get template resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.

Table 140. HTTP error response codes for a get template resource pool request (continued)

HTTP error status code	Description
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 141 on page 187](#).

Table 141. Response from a list template resource pool request

Field	Type	Description
rdp-list	Array	Array describing the resource pools. See “List template resource pools” on page 186 .

Table 142. Template resource pool list

Field	Type	Description
rdp-id	String	The generated ID for the resource pool.
rdp-name	String	Description name of the domain-shared resource pool. The resource pool name is in the form <i>domain-name.*</i> , where asterisks (*) are used to indicate that the resource pool is shared across the domain.
rdp-pool-type	String	Type of resource pool: rdp-dedicated Dedicated to a single software services template rdp-shared Shared amongst software services templates
rdp-shared-service-instance-snaapplid-name-prefix	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-subsystem-name-prefix	String	The software services instance name subsystem prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-shared-service-instance-general-name-prefix	String	The software services instance name general prefix for a shared resource pool. This property is returned only for shared resource pools.
rdp-domain-id	String	The generated ID for the domain to which the resource pool belongs.
rdp-tenant-id	String	The generated ID for the tenant to which the resource pool belongs.

Table 142. Template resource pool list (continued)

Field	Type	Description
rdp-template-name	String	Name of the software services template that the dedicated resource pool is associated with.
rdp-shared-template-name-list	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
rdp-instance-limit	Integer	Maximum number of software services instances that are allowed for the template.
rdp-instance-actual	Integer	Actual number of software services instances for the software services template that exist.
rdp-system-instance	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
rdp-user-instance-limit	Integer	Maximum number of software services instances that are allowed for a single user.
rdp-instance-expiration-limit	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
rdp-registry-instance-list	Array of Strings	Array of strings, where each string is the identifier of a registry instance that is associated with the resource pool.
rdp-system-pool	Array of JSON objects	Array that describes the systems selected for provisioning. See Table 128 on page 163 .
rdp-system-selection	String	Type of system selection for the resource pool: single Use a specific system. auto Assign a system automatically. prompt Prompt the user for the system.
rdp-ready	boolean	Indicates if the resource pool is ready for use. true The resource pool is ready for use. false The resource pool is not ready for use.
rdp-network-resources-needed	boolean	Indicates if a network resource pool is required. true A network resource pool is required. false A network resource pool is not required.
rdp-network-pool-id	String	Generated identifier of the network resource pool.

Table 142. Template resource pool list (continued)

Field	Type	Description
rdp-network-pool-ready	String	Indicates if the network resource pool is ready for use. complete The network resource pool is ready for use. incomplete The network resource pool is not yet ready for use. attention The network resource pool requires attention.
rdp-relocatable-resources	String	Optional, indicates if the resource pool can be relocated to a different system. movable The resource pool can be relocated to a different system. none Not specified.
rdp-sna-applid-prefixed-instance-name	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID. true The prefix should be derived from the SNA application ID. false The prefix should not be derived from the SNA application ID.
rdp-service-instance-name-prefix	String	The explicit prefix of the software services instance names.
rdp-wlm-resources-needed	boolean	Indicates if a workload management resource pool is required. true A workload management resource pool is required. false A workload management resource pool is not required.
rdp-wlm-sla	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
rdp-wlm-pool-id	String	Generated identifier of the workload management resource pool.
rdp-wlm-pool-ready	boolean	Indicates if the workload management resource pool is ready for use. complete The workload management resource pool is ready for use. incomplete The workload management resource pool is not yet ready for use.
rdp-report-class-name	String	The name of the workload management report class.

Table 142. Template resource pool list (continued)

Field	Type	Description
rdp-lpar-resources-needed	boolean	Indicates whether an LPAR resource pool is required. true An LPAR resource pool is required. false An LPAR resource pool is not required.
rdp-lpar-pool-ready	boolean	Indicates whether the LPAR resource pool is ready for use. complete LPAR resource pool is ready for use. incomplete LPAR resource pool is not yet ready for use.
rdp-job-statement	String	JOB statement JCL that is used in provisioning jobs.
rdp-share-tenant-instances	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool. true Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool. false Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.
rdp-account-modify	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. true The account information can be modified. false The account information cannot be modified .
create-time	String	Date and time that the resource pool was created.
created-by-user	String	User ID of the user that created the resource pool.
last-modified-time	String	Date and time that the resource pool was most recently modified.
last-modified-by-user	String	User ID of the user who last modified the resource pool.
object-uri	String	URI of the newly resource pool object.
local-system	Array	Array that describes the local system. See Table 128 on page 163 .

Table 142. Template resource pool list (continued)

Field	Type	Description
rdp-quietesced	boolean	Indicates if the resource pool is quietesced: true The resource pool is quietesced. You cannot provision any resources for this pool. false The resource pool is not quietesced. You can provision any resources for this pool.
rdp-tenant-report-class-name	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.
rdp-composite-cluster	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: true The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. false The resource pool is not to be used in a composite cluster.
rdp-cluster-name-prefix	String	The prefix used for cluster names.

Example HTTP interaction

In “List template resource pools” on page 186, a request is submitted to retrieve a resource pool.

```
GET https://localhost/zosmf/resource-mgmt/rest/1.0/domains/IYU1/template/template1/rdp
```

Figure 61. Sample request to list template resource pools

The following is the response body for the request to list domain resource pools in the previous example.

```
{
  "rdp-list": [
    {
      "rdp-id": "IYU10000",
      "rdp-name": "domain1.tenant1.template1",
      "rdp-pool-type": "rdp-dedicated",
      "rdp-quietesced": false,
      "rdp-domain-id": "IYU1",
      "rdp-tenant-id": "IYU100",
      "rdp-template-name": "template1",
      "rdp-instance-limit": 12,
      "rdp-instance-actual": 0,
      "rdp-system-instance": {
        "SY1": 0
      }
    }
  ]
}
```

```

    },
    "rdp-user-instance-limit": 0,
    "rdp-system-pool": [
      {
        "sysplex-name": "PLEX1",
        "sysplex-node-name": "SY1",
        "system-nickname": "SY1"
      }
    ],
    "rdp-system-selection": "single",
    "rdp-ready": true,
    "rdp-network-resources-needed": false,
    "rdp-network-pool-id": "",
    "rdp-network-pool-ready": "incomplete",
    "rdp-network-pool-local-ready": "incomplete",
    "rdp-sna-applid-prefixed-instance-name": false,
    "rdp-service-instance-name-prefix": "GP",
    "rdp-wlm-resources-needed": false,
    "rdp-wlm-sla": [],
    "rdp-wlm-pool-id": "",
    "rdp-wlm-pool-ready": "incomplete",
    "rdp-wlm-pool-local-ready": "incomplete",
    "rdp-job-statement": "",
    "rdp-account-modify": true,
    "rdp-share-tenant-instances": false,
    "rdp-composite-cluster": false,
    "rdp-cluster-name-prefix": null,
    "rdp-storage-resources-needed": false,
    "rdp-storage-pool": {
      "dataset-attributes-list": []
    },
    "rdp-instance-expiration-limit": 0,
    "rdp-lpar-resources-needed": false,
    "rdp-lpar-pool-ready": "incomplete",
    "create-time": "2022-01-07T14:18:28.616Z",
    "created-by-user": "ibmuser",
    "last-modified-time": "2022-01-07T14:18:28.629Z",
    "last-modified-by-user": "ibmuser",
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/IYU100000",
    "local-system": {
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SY1",
      "system-nickname": "SY1"
    },
    "provisioning-version": "1701",
    "provisioning-version-supported": true
  }
]
}

```

Update the security state for a tenant

Use this operation to update the tenant-state security field to a specified value for the specified tenant.

HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/state/actions/update
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
1.0.

<tenant-id>

Identifies the tenant.

Query parameters

None.

Description

This operation updates the tenant-state security field to the value that you specify in the request body. If the security definition is "Manual Security", use this API to set the state of the tenant to "Operational".

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the updating of the security state for the specified tenant.

Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 143 on page 193](#).

Table 143. Request content for the update security state request			
Field name	Type	Required or optional	Description
tenant-state	String	Required	The security state to assign to the tenant.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 144. HTTP error response codes for an update security state request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

Response content

None.

Example HTTP interaction

In [Figure 62 on page 194](#), a request is submitted to update the security state to "Operational" for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/state/actions/update
{
  "tenant-state": "Operational"
}
```

Figure 62. Sample request to update security state, with the request body

Get security resources

Use this operation to retrieve security profile information.

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/security-resources
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

Description

This operation retrieves security profile information.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in security profile information being retrieved.

Request content

None.

Authorization requirements

For more information, see [“Resource management services” on page 102](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned with a response body. See [“Response content” on page 195](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and the associated error message.

Table 145. HTTP error response codes for a get security resources request	
HTTP error status code	Description
HTTP 401 Unauthorized	The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the security profile. See [Table 146 on page 195](#).

Table 146. Response from a get security resources request		
Field	Type	Description
ServiceId	String	The ID of the service. For Cloud Provisioning, 5655S28PR00.
ServiceName	String	Descriptive name for the service. For Cloud Provisioning, IBM Cloud Provisioning and Management for z/OS.
MetaValidationItemVersion	Number	The meta validation item version. For Cloud Provisioning, 1.0.
Vendor	String	The vendor of the service. For Cloud Provisioning, IBM.
SecurityValidationItems	Array of Strings	The array of security validation items. There is 1 array entry for each security validation item. See Table 147 on page 195 .

Table 147. SecurityValidationItem structure		
Field	Type	Description
ItemId	String	The ID of the security validation item. For Cloud Provisioning, the first item is "5655S28PR00I00100000". Each subsequent item has the rightmost digit incremented by 1.
ItemType	String	The type of the item, PROGRAMMABLE, MANUAL or SEMI-PROGRAMMABLE.
ItemCategory	String	The category of the item.
ResourceProfile	String	The SAF resource profile. For Cloud Provisioning, this is the resource profile associated with the resource.
ResourceClass	String	The SAF resource class. For Cloud Provisioning, this is the resource class associated with the resource.
WhoNeedsAccess	String	The ID of the user/group needing access. When there are multiple IDs, they are separated by a space. For Cloud Provisioning, this is constructed based on the details of the resource.
LevelOfAccessRequired	String	The required access level. For Cloud Provisioning, this is the required access for the resource profile associated with the resource.

Table 147. <i>SecurityValidationItem</i> structure (continued)		
Field	Type	Description
ItemDescription	String	The description of the item. For Cloud Provisioning, this is constructed based on details of the resource.

Example HTTP interaction

In Figure 63 on page 196, a request is submitted to retrieve security profile information.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/security-resources
```

Figure 63. Sample request to get security profile information

The following is the response body for the example get security profile information request.

```
{
  "ServiceId": "5655S28PR00",
  "ServiceName": "IBM Cloud Provisioning and Management for z/OS",
  "MetaValidationItemVersion": 1,
  "Vendor": "IBM",
  "SecurityValidationItems": [
    {
      "ItemId": "5655S28PR00I00100012",
      "ItemType": "PROGRAMMABLE",
      "ItemCategory": "Cloud Provisioning Security Administrators",
      "ResourceProfile": "IZUDFLT.ZOSMF.SECURITY.ADMIN",
      "ResourceClass": "ZMFCLLOUD",
      "WhoNeedsAccess": "IZUSECAD",
      "LevelOfAccessRequired": "READ",
      "ItemDescription": "Grants the user the security administrator role."
    },
    {
      "ItemId": "5655S28PR00I00100006",
      "ItemType": "PROGRAMMABLE",
      "ItemCategory": "Cloud Provisioning z/OSMF",
      "ResourceProfile": "IZUDFLT.ZOSMF",
      "ResourceClass": "ZMFAPLA",
      "WhoNeedsAccess": "IZU0RPAW IYU IYU000 IYU0RPAW IYU0 PROVADM IZUADMIN ZOSMFAD
IZUUSER",
      "LevelOfAccessRequired": "READ",
      "ItemDescription": "Allows the user access to z/OSMF."
    }
  ]
}
```

Software services template services

The software services template services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services templates, which can be used to provision z/OS software in support of IBM Cloud Provisioning and Management for z/OS.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services” on page 46](#).

The basic procedure for provisioning software is:

1. Define domains and tenants. See [“Resource management services” on page 102](#).
2. Create a template, specifying the workflow, action and variables files that were provided by the software vendor.

The template is added to the software services catalog.

3. Add the template to a tenant.

4. Modify the template as needed.
5. Approve any approval records. Approval records are created when a workflow or action definition file contains an element that identifies a user ID under which a workflow step or action is to be performed (a runAsUser ID). They can also be defined for the template in general, and for a domain.
6. Test the template and ensure that it successfully creates an instance, that is, that it provisions the software and that the actions defined for the instance perform as expected. Optionally, clean up the results of your test, that is, deprovision and remove the instance that you created by testing the template.
7. Publish the template to make it available to consumers.
8. Run the template to create a software instance.

There are these types of templates:

Standard

Use these to provision a single software service. The preceding procedure assumes the use of standard templates.

Composite

Use these to provision more than one type of software service with a single Run operation. For more information, see [“Composite templates” on page 199](#).

Table 148 on page 197 lists the operations that the software services template services provide.

“Published software service template services” on page 283 describes the REST APIs for working with published software services templates, for example, for running a template to create an instance.

“Software services instance services” on page 319 describes the REST APIs for working with software services instances.

Software services template

Table 148. z/OSMF software services template services: operations summary	
Operation name	HTTP method and URI path
“Create a software services template” on page 203	POST /zosmf/provisioning/rest/<version>/scc
“Create a new version of a software services template” on page 211	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_new_version
“Create a new software services template based on an existing one” on page 218	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_based_on
“Modify a software services template” on page 221	POST /zosmf/provisioning/rest/<version>/scc/<object-id>
“Delete a software services template” on page 227	DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>
“List the software services templates” on page 253	GET /zosmf/provisioning/rest/<version>/scc

Table 148. z/OSMF software services template services: operations summary (continued)

Operation name	HTTP method and URI path
“Get a software services template” on page 228	GET /zosmf/provisioning/rest/<version>/scc/<object-id>
“Get a software services template history” on page 244	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/history
“Get software services template documentation” on page 246	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/admin GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/consumer
“Get prompt variables for a software services template” on page 247	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/prompt-variables
“Get source information for a software services template” on page 251	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/sources
“Publish a software services template” on page 259	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/publish
“Test a software services template” on page 261	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/test
“Refresh a software services template” on page 265	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/refresh
“Archive a software services template” on page 266	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/archive
“Add an approval for a software services template” on page 268	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
“Get an approval for a software services template” on page 269	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>
“List the approvals for a software services template” on page 272	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals

Table 148. z/OSMF software services template services: operations summary (continued)	
Operation name	HTTP method and URI path
“Approve an approval record for a software services template” on page 275	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>/actions/approve
“Batch approve approval records for a software services template” on page 276	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/actions/update
“Reject the use of a user ID with a software services template” on page 278	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>/actions/reject
“Delete an approval for a software services template” on page 279	DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>
“Set security complete for a software services template” on page 281	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/security_complete

Composite templates

Use a composite template to provision multiple related software services with a single Run operation. For example, you might use a composite template to provision CICS and z/OS Connect. A composite template contains other templates that are:

- Published
- Standard type. A composite template cannot contain other composite templates.

A composite template is associated with a specific domain. The published standard templates that it contains must be in that domain.

The standard templates that are members of a composite template dictate the sequence that they provisioned in.

Variables: A provider can satisfy prompt variables that are associated with the standard template using the connectors field. If a prompt variable is also specified as a connector variable, the prompting of that variable is automatically disabled, because it is satisfied through the connectors field.

The composite template can also take in an optional variable input file, the composite properties file. This file contains atCreate variable values that are associated with the member standard templates. It is an alternative to providing the atCreate values with the Run action. The atCreate variable names are in this format: <standard-template>.<atcreate-variable-name>. If the composite properties file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables associated with the member template. No validation is done on the values that are associated with the atCreate variables.

The precedence of values for the provisioning workflow is as follows. Values that are earlier in the list override values that are later in the list.

1. Connector and prompt values.
2. Values in the composite properties file.

The precedence of values for the action workflow is as follows:

1. Prompt values.
2. wfVar values that are specified in the actions definition.
3. Values in the composite properties file.

Resource pools: Like standard templates, composite templates must be associated with a tenant prior to being test run and run. The following describes values for the resource pools of a composite template:

instance name prefix

Specified by the resource pool for the composite template.

maximum number of instances

Specified by the resource pool for the composite template. It cannot exceed the smallest maximum of all of the standard template resource pools.

system selection

Specified by the resource pool for the composite template. The system selection is limited to the common systems that are referenced by the resource pools of standard templates that are associated with the composite template. All of the standard templates that are associated with the composite template are provisioned on the same system.

account information

Obtained from the resource pool that is associated with the standard template.

network resource pool

Not specified by the resource pool for the composite template.

workload management resource pool

Not specified by the resource pool for the composite template.

The resource pools that are associated with the standard templates that are referenced by the composite template must exist in the same tenant as the composite template.

Software services instances: When you use the Run operation for a composite template, multiple catalog type registry instances are created, one parent and a child for each standard template in the sequence.

The composite resource pool prefix is applied to the parent software services instance only. The standard template resource pool prefix is applied to each child software services instance.

An instance count is updated for both the composite resource pool and for each of the standard template resource pools.

The parent software services instance contains an array of composite registry objects, and each child includes the parent registry instance object ID.

Once all of the child software services instances are provisioned, the parent software services instance moves to the provisioned state, and you can use the child software services instances, that is, you can perform actions against them. The deprovisioning action is allowed only against the parent instance. The deprovisioning sequence is the opposite of the provisioning sequence.

If any of the children fail provisioning, you can either:

- Deprovision the failed provisioning child along with any child instances that have already been provisioned. Any child in the being-initialized state will remain as is – no deprovision action is run against it.
- Restart the failed child instance. If the restart is successful, it resumes the provisioning of the remaining children instances.

Once you have deprovisioned the parent instance (by using the **Perform deprovision** action against it), you can delete the parent instance, which also deletes all of the child instances.

Template Versions: When a new version of a standard template that is included in a composite template is published, any composite template that includes the standard template as a member is archived. The user then has the option to either re-publish one or more of the affected composite templates or create a new version of them.

When a standard template that is a member of one or more composite templates is moved out of published state (with the Archive or Delete actions) and a new standard template is not provided simultaneously, all affected composite templates are put into `missing_required_member` state. The composite templates remain in that state until a version of the missing member is published. The new version must be a version of the original member that was included in the composite definition. Once the missing member template is in publish state, the composite template is put into archive state if only that member template was missing. Otherwise, the composite template remains in `missing_required_member` state until all of the member templates are present. From the archive state, the provider or user can choose to re-publish the archived composite templates if the content of the standard templates and the connector information is still valid. If the content of the standard templates and the connector information is no longer valid, the user can create a new version of the archived composite template. The user should delete the previous version if it is no longer needed.

When all versions of a member template are deleted and a new unrelated standard template is published, all affected composite templates are put into `missing_required_member` state. The composite templates remain in that state indefinitely because there are no versions of the missing member template, and so the requirement that the member must be a version of the original member of the composite definition cannot be satisfied. The user can either delete the composite template or create a new version of it.

Usage scenario: Two published templates, `template1` and `template2`, are located in the same domain, and are associated with the same tenant, with at least one system in common.

1. A provider creates a composite template from the published standard templates, specifying `template1` as sequence 1, and `template2` as sequence 2, with a connector value, `TEMP2_VAR1 = TEMP1_VAR1` from `template1`.
2. The provider associates the composite template with the tenant, creates the resource pool, and then test runs the template.
3. The provider displays the instances table in the Software Services task. After the parent instance is in a provisioned state, the provider performs actions against the child instance for `template1`.
4. When the instance is no longer needed, the provider uses an action to deprovision the parent instance.
5. Once the parent instance is in a deprovisioned state, the provider removes it. This also removes all of the child instances.

Clustered composite templates

Clustered composite templates allow you to leverage sysplex capabilities to provision a continuously available middleware environment. With a single provisioning action, you provision network-clustered instances of a specific middleware in a sysplex. Similarly, a single deprovision action releases all of the member instances that are associated with the clustered composite template instance.

You create a clustered composite template from a single published template or from multiple published templates that use the **Use the composite template to cluster instances on systems in a sysplex** option when adding a template. The published templates must all be of the same software type (that is, they provision the same middleware).

Provisioning a clustered composite template results in each instance of the member templates being provisioned on a separate system. As a result, the total number of instances defined in a clustered composite template is limited, based on several factors, including whether the composite template resides in a single sysplex domain or a multiple sysplex domain. In a single sysplex domain, the total number of instances cannot exceed the number of systems in the domain or the number of systems in any of the resource pools that are associated with the clustered composite template definition. In a multiple-sysplex domain, the maximum number is based on the sysplex that contains the most systems

in the domain; the instances will be created in this sysplex. As an example, assume that a domain encompasses systems on two sysplexes: System 1 on Sysplex A and Systems 2 and 3 on Sysplex B. Here, the maximum number of clustered instances that can be created is two because Sysplex B has two systems in the domain.

Clustered composite templates have their own resource pools. z/OS resources for all of the member instances are obtained from the same resource pool when the clustered composite template is provisioned. All of the systems in the resource pool must be a member of the same sysplex.

Authorization requirements

Use of the software services template services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the provisioning administrator and domain administrator. The specific requirements for each software services template service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements”](#) on page 49. For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 149. Response from a software services template request failure		
Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

Error logging

Errors from the software services template services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in [IBM z/OS Management Facility Configuration Guide](#).

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request succeeded and resulted in the creation of an object.

HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

HTTP 204 No content

The request succeeded, but no content is available to be returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 404 Not found

The requested resource does not exist.

HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

Related information

The publish operation locks the template, preventing any further modification, and exposes the template to consumers. To work with a published software services template, use the REST APIs that are described in [“Published software service template services”](#) on page 283.

The run operation for a published template creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry. To work with a software services instance, use the REST APIs described in [“Software services instance services”](#) on page 319.

Create a software services template

Use this operation to create a software services template in the catalog. The template is a private entry until it is published.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates a software services template in the catalog, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 203.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new software services template. A response body is provided, as described in [“Response content”](#) on page 208.

Request content

The request content is expected to contain a JSON object that describes the software services template to be created. See [Table 150](#) on page 204.

Table 150. Request content for a request to create a software services template.

Field name	Type	Required or optional	Valid for Template Type	Description
template-type	String	Optional	Standard, Composite	Identifies the type of template: standard Defines a single software service. composite Consists of multiple published templates that are provisioned together. If template-type is not specified, the type defaults to standard.
composite-cluster	Boolean	Optional	Composite	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
composite-definition	Array of objects	Required	Composite	An array of objects that define the composite template. See Table 151 on page 207 .
action-definition-file	String	Required	Standard	Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml. File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows. Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name. Partitioned data set <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
description	String	Optional	Standard, Composite	Description of the software services template.

Table 150. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
name	String	Required	Standard, Composite	Descriptive name for the software services template. The name must be unique, no longer than 48 characters, and consist of alphanumeric characters (A-Z, a-z, and 0-9), national characters (\$@), underscore (_), and hyphen (-).
workflow-definition-file	String	Required	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p>Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p>Partitioned data set</p> <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
workflow-variable-input-file	String	Optional	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 150. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
workflows-disposition	String	Optional	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete. The default is archive. If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.
composite-variable-input-file	String	Optional	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties The variable names are in the following format: <standard-template-name>.<atcreate-variable> For example: CICS.startup=10 If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.
jobs-disposition	String	Optional	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete. The default is keep.
instances-disposition	String	Optional	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete. The default is keep.
domain-name	String	Optional	Standard, Composite	Name of the domain. Required if the user ID has administrator authorization to more than one domain.
approvals	Array of strings	Optional	Standard, Composite	An array of strings representing user IDs of users that are responsible for approving the template.

Table 150. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
workflow-clean-after-provisioned	boolean	Optional	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.
consumer-documentation-file	String	Optional	Standard, Composite	Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
consumer-documentation-type	String	Optional	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
admin-documentation-file	String	Optional	Standard, Composite	Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
admin-documentation-type	String	Optional	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.

Table 151. Composite-definition structure

Field	Type	Required/ optional	Description
sequence	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of objects	Optional	An array of connector object. Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above. See Table 152 on page 208.

Table 152. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 208](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 153 on page 208](#) lists the fields in the JSON object.

Table 153. Response from a create software services template request

Field	Type	Description
generated-name	String	The generated name associated with this software services template.
object-id	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
object-uri	String	The object URI of the newly created software services template.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 154 on page 209 .

Table 154. Response from a create request: SAF-resource object		
Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 155. Response from a software services template request failure		
Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

Example HTTP interaction

The example in [Figure 64 on page 210](#) shows a request to create a standard software services template on the system SY1.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc

{
  "name": "s3Suspend",
  "action-definition-file": "/u/wfSuspend/a.xml",
  "workflow-definition-file": "/u/wfSuspend/p.xml",
  "domain-name": "default",
  "workflows-disposition": "keep",
  "jobs-disposition": "keep",
  "description": "This service suspends in the midst of provisioning and deprovisioning.",
  "approvals": ["zosmfad"]
}
```

Figure 64. Sample request to create a standard software services template

The following is the response body for the request.

```
{
  "generated-name": "s3Suspend.1.default",
  "object-id": "5e3c224b-eb47-47f9-847f-89456850f8aa",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/5e3c224b-eb47-47f9-847f-89456850f8aa",
  "SAF-resources": [
    {
      "description": "Controls which users are template approvers for the s3Suspend template in the default domain.",
      "ids": [
        "zosmfad"
      ],
      "groups": [],
      "role": "Template Approver",
      "resource-class": "ZMFCLLOUD",
      "resource-name": "IZUDFLT.ZOSMF.TEMPLATE.APPROVERS.IYU0.s3Suspend",
      "required-access": "SAF_READ",
      "other-required-ids": [],
      "audit-requirements": ""
    }
  ]
}
```

Figure 65. Sample response body

The example in [Figure 66 on page 210](#) shows a request to create a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc

{
  "domain-name": "default",
  "name": "S1_S2",
  "template-type": "composite",
  "composite-cluster": false,
  "description": "This is a composite template that brings up services s1 and s2. The s2 services references information from the s1 services to satisfy its run-time properties.",
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "INS",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }, {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "WELSHIE"
        }
      ]
    }
  ]
}
```

Figure 66. Sample request to create a composite software services template

The following is the response body for the request.


```
{
  "generated-name": "S1_S2.1.default",
  "object-id": "5f746dfc-ad24-4355-99d3-b83466ce4492",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/5f746dfc-ad24-4355-99d3-b83466ce4492",
  "SAF-resources": []
}
```

Figure 67. Sample response body for a composite template

Create a new version of a software services template

You can use this operation to create a new version of a software services template, with the same name as the original, associated with the same domain and tenants, but with new source files.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_new_version
```

In this request

<object-id>

Identifies the existing software services template to create a new version of.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates a new version of an existing software services template in the catalog. The new version has the same name as the original entry, and is associated with the same domain and tenants. However, it has new source files (workflow definition, action definition, variable input, and documentation). You cannot already have a draft software services template of this version.

The new version is assigned a version number that is the next available number in sequence.

The template that you create a new version of must be in the published, archived, or missing member state.

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the request resulted in the creation of a new version of a software services template. A response body is provided, as described in [“Response content” on page 216](#).

Request content

The request content is expected to contain a JSON object that describes the software services template to be created. See [Table 156 on page 212](#).

Table 156. Request content for the software services template request

Field name	Type	Required or optional	Valid for Template Type	Description
action-definition-file	String	Required	Standard	<p>Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p>Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p>Partitioned data set</p> <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
composite-definition	Array of objects	Required	Composite	<p>An array of objects that define the composite template.</p> <p>See Table 157 on page 215.</p>
composite-cluster	Boolean	Optional	Composite	<p>Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.</p>
description	String	Optional	Standard, Composite	<p>Description of the software services template, up to 500 characters.</p>

Table 156. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
workflow-definition-file	String	Required	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf//samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p>Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p>Partitioned data set</p> <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
workflow-variable-input-file	String	Optional	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 156. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
composite-variable-input-file	String	Optional	Composite	<p>Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties</p> <p>The variable names are in the following format: <standard-template-name>.<atcreate-variable></p> <p>For example: CICS.startup=10</p> <p>If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.</p>
approvals	Array of strings	Optional	Standard, Composite	An array of strings representing user IDs of users that are responsible for approving the template.
workflow-clean-after-provisioned	Boolean	Optional	Standard	This field is ignored. The workflows-disposition field should be referenced instead. If the workflows-disposition field is not provided, its default value of archive is used.
consumer-documentation-file	String	Optional	Standard, Composite	Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
consumer-documentation-type	String	Optional	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
admin-documentation-file	String	Optional	Standard, Composite	Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
admin-documentation-type	String	Optional	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.

Table 156. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
workflows-disposition	String	Optional	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete. The default is archive. If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.
jobs-disposition	String	Optional	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete. The default is keep.
instances-disposition	String	Optional	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete. The default is keep.

Table 157. Composite-definition structure

Field	Type	Required/ optional	Description
sequence	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of objects	Optional	An array of connector object. Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above. See Table 158 on page 216 .

Table 158. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 216](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 159 on page 216](#) lists the fields in the JSON object.

Table 159. Response from a create new version of a software services template request

Field	Type	Description
generated-name	String	The generated name associated with this software services template.
object-id	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
object-uri	String	The object URI of the newly created software services template.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 160 on page 217 .

Table 160. Response from a create request: SAF-resource object

Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

Example HTTP interaction

In Figure 68 on page 217, a request is submitted to create a new version of a software services template on the system SY1.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463/actions/create_new_version
{
  "workflow-definition-file":"/u/wfSuspend/p.xml",
  "action-definition-file":"/u/wfSuspend/a.xml",
  "approvals":["zosmft1", "zosmfad"]
}
```

Figure 68. Sample request to create a new version of a software services template

```
{
  "generated-name": "s2.3.default",
  "object-id": "3f8ca645-f872-42b6-b0fc-3c6a9e470fcc",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/3f8ca645-f872-42b6-b0fc-3c6a9e470fcc",
  "SAF-resources": [
    {
      "description": "Controls which users are template approvers for the s2 template in the default domain.",
      "ids": [
        "zosmfad",
        "zosmft1"
      ],
      "groups": [],
      "role": "Template Approver",
      "resource-class": "ZMFCLLOUD",
      "resource-name": "IZUDFLT.ZOSMF.TEMPLATE.APPROVERS.IYU0.s2",
      "required-access": "SAF_READ",
      "other-required-ids": [],
      "audit-requirements": ""
    }
  ]
}
```

Figure 69. Sample response body

The example in Figure 66 on page 210 shows a request to create a new version of a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/5f746dfc-ad24-4355-99d3-b83466ce4492/
actions/create_new_version

{
  "description": "This is an updated version of a composite template that brings up services s1 and s2.",
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }
      ]
    }
  ]
}
```

Figure 70. Sample request to create a new version of a composite software services template

The following is the response body for the request.

```
{
  "generated-name": "S1_S2.2.default",
  "object-id": "6f72b8d9-26ae-4552-9f4b-11eadff2225e",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/6f72b8d9-26ae-4552-9f4b-11eadff2225e",
  "SAF-resources": []
}
```

Create a new software services template based on an existing one

You can use this operation to create a new software services template based on one that already exists, with the same source files. This operation is not valid for composite templates.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_based_on
```


In this request

<object-id>

Identifies the existing software services template.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates a new software services template in the catalog, based on the existing software services template identified by the object ID. It has the same source files (workflow definition, action definition, variable input, and documentation).

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the request resulted in the creation of a new version of a software services template. A response body is provided, as described in [“Response content” on page 220](#).

Request content

The request content is expected to contain a JSON object. See [Table 161 on page 219](#).

Table 161. Request content for the software services template request			
Field name	Type	Required or optional	Description
name	String	Required	Descriptive name for the software services template. The name must be unique, no longer than 48 characters, and consist of alphanumeric characters (A-Z, a-z, and 0-9), national characters (\$@), underscore (_), and hyphen (-).
domain-name	String	Varies	Name of the domain. Required if the user ID has administrator privileges to more than one domain.
approvals	Array of strings	Optional	An array of strings representing the user IDs that are responsible for approving the template.
target-copy-path	String	Required	The absolute path name of an empty z/OS UNIX directory. The source file contents of the existing software services template are copied into this location, and the new template is created based on that content. If the directory does not exist, it is created. However, the parent directory must already exist.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 220](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 162 on page 220](#) lists the fields in the JSON object.

Table 162. Response from a successful request		
Field	Type	Description
generated-name	String	The generated name associated with this software services template.
object-id	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
object-uri	String	The object URI of the newly created software services template.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 163 on page 220 .

Table 163. Response from a create request: SAF-resource object		
Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

Example HTTP interaction

In [Figure 71 on page 221](#), a request is submitted to create a new version of a software services template on the system SY1.

```
POST /zosmf/provisioning/rest/1.0/scc/0389ed37-fe13-4176-af65-c171b6ba6b37/actions/create_based_on
HTTP/1.1

{
  "name" : "config2",
  "target-copy-path": "/users/gg/zosmf/newConfig2"
}
```

Figure 71. Sample request to create a new software services template based on an existing one, with request body

```
{
  "generated-name": "mqUpgrade.1.default"
  "object-id": "cd00fb41-20ed-4133-b985-52e28edfcfd0"
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/cd00fb41-20ed-4133-b985-52e28edfcfd0",
  "SAF-resource": []
}
```

Figure 72. Sample response body

Modify a software services template

You can use this operation to modify fields in a software services template in the catalog.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request

<object-id>

Identifies the software services template to be modified.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation modifies fields in a software services template in the catalog, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 221.

On successful completion, HTTP status code 204 (Normal) is returned, indicating that the request resulted in a modified software services template.

The software services template must be in one of the draft states.

Modifying any of the definition files causes all approvals to be reset.

Request content

The request content is expected to contain a JSON object that describes the fields to be modified. See [Request content for the software services template request](#).

Table 164. Request content for a request to modify a software services template

Field name	Type	Valid for Template Type	Description
composite-definition	Array of objects	Composite	An array of objects that define the composite template. See Table 165 on page 225 .
composite-cluster	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
action-definition-file	String	Standard	<p>Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p>Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p>Partitioned data set</p> <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
description	String	Standard, Composite	Description of the software services template.

Table 164. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
workflow-definition-file	String	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf//samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p>Sequential data set Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p>Partitioned data set</p> <ul style="list-style-type: none"> • If the action definition file is in the same data set as the file template: just the member name. • If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.
workflow-variable-input-file	String	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 164. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
composite-variable-input-file	String	Composite	<p>Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties</p> <p>The variable names are in the following format: <standard-template-name>.<atcreate-variable></p> <p>For example: CICS.startup=10</p> <p>If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.</p>
workflow-clean-after-provisioned	Boolean	Standard	<p>This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.</p>
consumer-documentation-file	String	Standard, Composite	<p>Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>
consumer-documentation-type	String	Standard, Composite	<p>Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified with a value that is not null.</p>
admin-documentation-file	String	Standard, Composite	<p>Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>
admin-documentation-type	String	Standard, Composite	<p>Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified with a value that is not null.</p>

Table 164. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
approvals	Array of strings	Standard, Composite	General approvals that are associated with the template. Each string represents the user ID of a general approval. If the array contains a user ID for a general approval that already exists for the template, the status and all of the corresponding information for that user ID is maintained for the template. If the array does not contain a user ID for a general approval that already exists for the template, that user ID is removed from the general approval list for the template. An empty array removes any existing general approvals from the template. A null value for approvals results in no changes.
workflows-disposition	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete. The default is archive. If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.
jobs-disposition	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete. The default is keep.
instances-disposition	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.

Table 165. Composite-definition structure

Field	Type	Required/ optional	Description
sequence	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of objects	Optional	An array of connector object. Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above. See “Modify a software services template” on page 221 .

Table 166. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Example HTTP interaction

[Figure 73 on page 226](#) shows a request to modify a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/6e39d092-745a-4e81-8b7b-d3e1327ad230
{
  "workflow-variable-input-file":"/u/wfStandard/p.props"
}
```

Figure 73. Sample request to modify a software services template

[Figure 74 on page 227](#) shows a request to modify a composite software services template.

POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8

```
{
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        },
        {
          "variable-name": "INS",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }
      ]
    }
  ]
}
```

Figure 74. Sample request to modify a composite software services template

Delete a software services template

You can use this operation to delete a software services template from the catalog.

HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request

<object-id>

Identifies the software services template to be deleted.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation deletes a software services template from the catalog.

On successful completion, HTTP status code 200 Ok is returned, indicating that the request resulted in a software services template being deleted.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 Ok is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 167 on page 228](#) lists the fields in the JSON object.

Table 167. Response from the software services template request		
Field	Type	Description
composites-affected	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 168. Composite-affected object		
Field	Type	Description
name	String	Name of the composite template that was affected by an update to a standard template.
state	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

Example HTTP interaction

[Figure 75 on page 228](#) shows a request to delete a software services template.

```
DELETE https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/33d4171f-b759-4926-8a48-e359e589474a
```

Figure 75. Sample request to delete a software services template

The response body is as follows.

```
{
  "composites-affected": null
}
```

Get a software services template

Use this operation to retrieve a software services template from the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template to retrieve.

Query parameters

None.

Description

This operation retrieves a software services template from the catalog.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 229](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 169 on page 229](#) lists the fields in the JSON object.

Table 169. Response from a request to get a software services template.			
Field	Type	Valid for Template Type	Description
base-object-id	String	Standard, Composite	The object ID that is associated with all of the versions of the software services template.
generated-name	String	Standard, Composite	Generated name for the software services template.
name	String	Standard, Composite	The name associated with the software services template.
version	String	Standard, Composite	Version of the software services template.
owner	String	Standard, Composite	User ID of the software services template owner.

Table 169. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
state	String	Standard, Composite	Indicates the status of the software services template. See “State values” on page 234.
description	String	Standard, Composite	Description of the software services template.
tenants	Array of Strings	Standard, Composite	Each string represents a tenant that the template is associated with.
domain-shared-tenants	Array of Strings	Standard, Composite	Each string represents a tenant in the domain that the template is associated with through the domain shared resource pool.
domain-name	String	Standard, Composite	The domain the template is associated with.
approvals	Array of objects	Standard, Composite	Array of Approval-Object containing information about the approvals associated with this software services template. See Table 170 on page 235.
action-definition-file	String	Standard	Location of the action definition file.
action-definition-file-original-source	String	Standard	Original user specified location of the action definition file
action-definition-file-original-timestamp	String	Standard	Last-modified time stamp for when the original action definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
actions	Array of objects	Standard	Array of Action-Object containing information about the actions associated with the template. See Table 171 on page 236.
software-id	String	Standard	A short, arbitrary, value that identifies the software that is being provisioned.
software-name	String	Standard	Name of the software that is being provisioned.
software-type	String	Standard	Type of software that is being provisioned.
software-version	String	Standard	Version of the software that is being provisioned.
workflow-definition-file	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow
workflow-definition-file-original-source	String	Standard	Original user-specified location of the workflow definition file.

Table 169. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
workflow-definition-file-original-timestamp	String	Standard	The last-modified time stamp for when the original workflow definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-id	String	Standard	A short, arbitrary value that identifies the workflow.
workflow-vendor	String	Standard	Name of the vendor that provided the workflow definition file.
workflow-version	String	Standard	Version of the workflow definition file.
composite-variable-input-file	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files. The variable names are in the following format: <standard-template-name>.<atcreate-variable> For example: CICS.startup=10
composite-variable-input-file-original-source	String	Composite	Location of the composite variable input file, an optional properties file used to specify in advance the values for one or more of the atCreate variables.
composite-variable-input-file-original-timestamp	String	Composite	The last-modified time stamp for when the original composite variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-variable-input-file	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
workflow-variable-input-file-original-source	String	Standard	The original user-specified location of the workflow variable input file.
workflow-variable-input-file-original-timestamp	String	Standard	The last-modified time stamp for when the original variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-clean-after-provisioned	Boolean	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.

Table 169. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
prompt-variables	Array of objects	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See Table 173 on page 237.
public-variables	Array of strings	Standard	Array of strings that name the public variables for the template.
at-create-variables	Array of strings	Standard, Composite	<p>Array of strings that name the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both.</p> <p>For a composite type template, any atCreate variable that was designated as a connector variable is excluded from the list. The variables are prefixed by the standard template name, for example MQ.BRING_UP.</p>
consumer-documentation-file	String	Standard, Composite	Location of the original file that provides information for consumers about the template.
consumer-documentation-file-original-source	String	Standard, Composite	The original user-specified location of the consumer documentation file.
consumer-documentation-type	String	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
admin-documentation-file	String	Standard, Composite	Location of a file that provides information for administrators about the template.
admin-documentation-file-original-source	String	Standard, Composite	The original user-specified location of the admin documentation file.
admin-documentation-type	String	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.
create-time	String	Standard, Composite	Time that this object was created, in ISO 8601 format.
create-by-user	String	Standard, Composite	User who created this object.
last-modified-time	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.

Table 169. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
last-modified-by-user	String	Standard, Composite	User who last updated this object.
published-timestamp	String	Standard, Composite	<p>The last time this template was moved to published state. If the template was never in the published state, an empty string is returned. ISO 8601 format.</p> <p>When you upgrade to the April, 2017 deliverable of z/OSMF, the published-timestamp for a template already in a published state is set to the timestamp of the upgrade.</p>
archived-timestamp	String	Standard, Composite	<p>The last time this template was moved to the archived state. If the template was never in the archived state, an empty string is returned. ISO 8601 format.</p> <p>When you upgrade to the April, 2017 deliverable of z/OSMF, the archived-timestamp for a template already in an archived state is set to the timestamp of the upgrade.</p>
workflows-disposition	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.
jobs-disposition	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.
instances-disposition	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.
automatic-security	String	Standard, Composite	<p>Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization:</p> <ul style="list-style-type: none"> • true if the domain that this template is associated with is set up for automatic authorization • false if the domain that this template is associated with is set up for manual authorization.
SAF-resources	Array of objects	Standard, Composite	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 174 on page 238 .
runAsUsers	Array of objects	Standard	Array of RunAsUser objects containing information about runAsUser IDs that are referenced by this template. See Table 175 on page 238 .

Table 169. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
runAsUser-audit	boolean	Standard	<p>Indicates if auditing is performed on workflows and action commands that are associated with the template. This field cannot be updated and is based on the level of the Cloud Provisioning plug-in at the time that the template is created.</p> <p>false runAsUser auditing is performed. This value is used for all templates created prior to the April, 2017 delivery.</p> <p>true runAsUser auditing is not performed. This value is used for all templates created beginning with the April, 2017 delivery.</p>
template-type	String	Standard, Composite	<p>Identifies the type of template.</p> <p>standard Defines a single software service.</p> <p>composite Consists of multiple published templates that will be provisioned together.</p>
composite-cluster	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
composite-definition	Array of objects	Composite	An array of objects that define the composite template. See Table 176 on page 239 .
composite-parents	Array of strings	Standard	<p>An array of strings. Each string is a composite template that includes this standard template. For example:</p> <pre>[c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]</pre>
security-wf-info	Object	Standard, Composite	<p>An object that describes the security workflow, when automatic security is in effect and an attempt to grant authorization to a runAsUser ID or approver ID failed.</p> <p>See Table 178 on page 241.</p>
provisioning-version	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.

State values

archived

The entry is hidden from consumers. You can make it available again with the **Publish** action.

corrupted

The contents of the software services template are missing or incorrect. Delete the template.

draft

The entry is in the edit state and visible only to the owner and the administrator. No approvals are required for this template. The entry can be tested with the Test a software services template API.

draft_approved

The software services template is in edit state and all the approvals that are associated with the template and the respective runAs user IDs have been received. The entry can be tested with the Test a software services template API.

draft_pending_approvals

The software services template is in edit state and one or more associated approvals has not been approved. The entry cannot be tested (with the Test a software services template API) in this state.

draft_missing_required_approver

One or more of the definition files contains a runAsUser element without a corresponding approver element. Either an approver element must be added for the runAsUser element, or a domain or general approver must be added for the software services template. The entry cannot be tested (with the Test a software services template API) in this state.

draft_rejected

The template is in a draft state and one or more approvers rejected an approval. The entry cannot be tested (with the Test a software services template API) in this state.

missing_required_member

One or more of the members of a composite template that was referenced in the composite definition is not available. This state applies only to published or archived composite templates.

pending_security_update

Permission to access the software services template is being processed. No API requests are allowed for the software services template until the security processing is complete.

published

The entry is locked and visible to consumers.

security_update_failed

Security access setup related to the software services template failed. Only the view and delete API requests are available.

Table 170. Response from a get request: Approval-Object		
Field	Type	Description
status	String	Status of the approval for this object: pending, approved, or rejected.
comment	String	Comment associated with the change in status from pending to either approved or rejected.
description	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition, for example, This workflow definition is associated with the <action-name> action.
approvers	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
status-update-by	String	User ID that performed the last approve or reject action for this approval object.
time-of-update	String	The last time this object was updated, in ISO 8601 format.
run-as-user	String	The runAsUser user ID that the approval object is for. Only applicable when the type is action_definition or step_definition.

Table 170. Response from a get request: Approval-Object (continued)

Field	Type	Description
type	String	Type of approval object: general, domain, action_definition, or step_definition.
object-id	String	Unique object ID representing this approval object.
workflow-file	String	Workflow file definition associated with this runAsUser user ID.
variable-input-file	String	Variable input file associated with this runAsUser user ID.
step-name	String	Workflow file definition step associated with this runAsUser user ID.
called-by-step-name	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
called-by-workflow-file	String	Workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
actions-file	String	Actions file definition associated with this runAsUser user ID.
action-name	String	Action that is defined in the actions file associated with this runAsUser user ID.
run-as-user-dynamic	boolean	Indicates if the run-as-user ID value can change: true The run-as-user ID value is not final and can change during the processing of the workflow false The run-as-user ID is final and cannot change during the processing of the workflow.

Table 171. Response from a get request: Action-Object

Field	Type
name	String
type	String
is-deprovision	String. The value must be either true or false.
command	String
command-run-as-user	String
command-sol-key	String
command-unsol-key	String
command-detect-time	String
workflow-definition-file	String
workflow-variable-input-file	String
workflow-variables	Variable[]
instructions	String

Table 171. Response from a get request: Action-Object (continued)

Field	Type
prompt-variables	String. The prompt variable objects that are associated with the action.
workflow-clean-after-complete	String. The value must be true, false, or inherit. For workflow type actions, if provided, this indicates whether the workflow instance is cleaned up after completion. If workflow-clean-after-complete is not provided, the default is inherit, and the value is inherited from the workflows-disposition field.
command-run-as-user-dynamic	boolean. Indicates if the command-run-as-user ID value can change: true The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content. false The command-run-as-user ID is final and cannot change during the processing of the command.

Table 172. Response from a get request: Variable-Object

Field	Type
name	String
value	String
visibility: public or private	String

Table 173. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.

Table 173. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Fields of type String default to null.

Table 174. Response from a create request: SAF-resource object

Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Table 175. Response from a get request: RunAsUser object

Field	Type	Description
description	String	Additional detail provided if the run-as-user is for a workflow definition that is associated with the action definition. Example: This workflow definition is associated with the <action-name> action.
approver-user-ids	Array of Strings	Array of strings where each string is a user ID that originates from the approver element that is associated with the runAsUser for the template step or action.

Table 175. Response from a get request: RunAsUser object (continued)		
Field	Type	Description
run-as-user	String	The runAsUser user ID that the approval object is for. This is applicable only to action_definition and step_definition type.
type	String	One of the following: action_definition or step_definition
workflow-file	String	The workflow file definition that is associated with this runAsUser user ID.
variable-input-file	String	The variable input file that is associated with this runAsUser user ID.
step-name	String	The workflow file definition step that is associated with this runAsUser user ID.
called-by-step-name	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the step in the parent workflow definition that called the workflow definition file that generated the approval object.
called-by-workflow-file	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the workflow definition that called the workflow definition file that generated the approval object.
actions-file	String	The actions file definition that is associated with this runAsUser user ID.
action-name	String	The action defined in the actions file that is associated with this runAsUser user ID.
run-as-user-dynamic	boolean	Indicates if the run-as-user ID value can change: true The run-as-user ID value is not final and can change during the processing of the workflow false The run-as-user ID is final and cannot change during the processing of the workflow.

Table 176. Response from a get request: Composite-definition object		
Field	Type	Description
sequence	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Indicates the number of child instances to be created using the template in a composite cluster. This field is within each entry of the composite-definition.
published-template-name	String	The name of an existing published template in the domain that is associated with the composite template.

Table 176. Response from a get request: Composite-definition object (continued)

Field	Type	Description
connectors	Array of objects	<p>An array of connector object.</p> <p>Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above.</p> <p>See Table 177 on page 240.</p>
prompt-variables	Array of objects	<p>Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template.</p> <p>If specified, this overrides the array of prompt variables that are associated with the template specified with published-template-name. Only prompt variables that are already specified for the published-template-name can be specified. An empty array will translate into not prompting for any variables. If this field is not provided or set to null, then the prompt variables that are associated with published-template-name are used.</p> <p>If the connector variable-name is also a prompt-variable, then the connector takes precedence, and the variable is not promptable.</p>
missing	boolean	<ul style="list-style-type: none"> • true if no published template is available that is related to the original version used when the template was defined • false if a published template exists that satisfies the published template requirement

Table 177. Connector object

Field	Type	Required/optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

Table 177. Connector object (continued)

Field	Type	Required/ optional	Description
not-valid	boolean	Required	Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is: <ul style="list-style-type: none"> false, if all of the information is accurate true, if one or more of the values are incorrect.

Table 178. Response from a get request: Security Workflow Information

Field	Type	
wf-status	String	Status of the workflow
wf-uri	String	URI of the workflow
wf-key	String	Key of the workflow
additional-info	String	Additional information about the error

Example HTTP interaction

In Figure 76 on page 241, a request is submitted to retrieve a standard software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/5ccbad22-94fd-4b31-bb2b-95aa8602cc48
```

Figure 76. Sample request to retrieve a standard software services template

The following is the response body for the example GET request.

```
{
  "name": "mqCBA",
  "version": "1",
  "owner": "domadmin",
  "state": "published",
  "description": "This workflow provisions an MQ for z/OS Queue Manager",
  "tenants": [...],
  "actions": [...],
  "approvals": [],
  "tested": false,
  "generated-name": "mqCBA.1.default",
  "domain-name": "default",
  "action-definition-file": "definition/qmgrActions.xml",
  "action-definition-file-original-source": "/users/gg/mqCBA/definition/qmgrActions.xml",
  "action-definition-file-original-timestamp": "2016-11-18T20:00:42Z",
  "software-id": "5655-W97",
  "software-name": "IBM MQ for z/OS",
  "software-type": "QMGr",
  "software-version": "V8.0.0",
  "workflow-definition-file": "definition/provision.xml",
  "workflow-definition-file-original-source": "/users/gg/mqCBA/definition/provision.xml",
  "workflow-definition-file-original-timestamp": "2016-11-18T20:03:47Z",
  "workflow-id": "ProvisionQueueManager",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0.1",
  "workflow-variable-input-file": "definition/workflow_variables.properties",
  "workflow-variable-input-file-original-source":
    "/users/gg/mqCBA/definition/workflow_variables.properties",
  "workflow-variable-input-file-original-timestamp": "2016-11-18T20:00:42Z",
  "prompt-variables": [],
  "public-variables":
    ["CSQ_CHIN_SERVICE_CLASS_NAME", "CSQ_MSTR_SERVICE_CLASS_NAME", "CSQ_TCPIP_PORT_NUMBER",
     "CSQ_AUTO_GEN_CMD_PFX_SSID", "CSQ_CMD_PFX_FOR_AUTO_GEN", "CSQ_CHIN_REPORT_CLASS_NAME",
     "CSQ_MSTR_CLASSIFICATION_RULE_ID", "CSQ_MSTR_REPORT_CLASS_NAME", "CSQ_CMD_PFX", "CSQ_QSGDISP",
```

```

"CSQ_CHIN_CLASSIFICATION_RULE_ID", "CSQ_TCPIP_STATUS_CODE", "CSQ_TARG_LIB_HLQ", "CSQ_SSID",
"CSQ_TCPIP_PORT_ID", "CSQ_LANG_LETTER", "CSQ_ENVIRONMENT"
]

"at-create-variables": [],
"workflow-clean-after-provisioned": true,
"security-wf-info": null,
"create-time": "2016-11-18T20:00:43.504Z",
"created-by-user": "domadmin",
"last-modified-by-user": "domadmin",
"last-modified-time": "2016-11-18T20:04:50.913Z",
"admin-documentation-file-original-source": "/users/gg/mqCBA/documentation/admin-
mqaas_readme.pdf",
"admin-documentation":
"/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
admin",
"admin-documentation-type": "pdf",
"consumer-documentation-file-original-source":
"/users/gg/mqCBA/documentation/consumer-workflow_variables.properties",
"consumer-documentation":
"/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
consumer",
"consumer-documentation-type": "text",
"base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
"admin-documentation-mime-type": "application/pdf",
"consumer-documentation-mime-type": "text/plain",
"SAF-resources": [],
"runAsUsers": [],
"runAsUser-audit": true,
"automatic-security": true,
"published-timestamp": "2017-04-05T16:16:55.878Z",
"archived-timestamp": "",
"provisioning-version": "1400"
}

```

Figure 77 on page 242 shows a request to retrieve a composite software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463
```

Figure 77. Sample request to retrieve a composite software services template

The following is the response body for the GET request for the composite software services template.

```

{
  "name": "S1_S2",
  "owner": "ibmuser",
  "state": "missing_required_member",
  "description": "This is a composite template that brings up services s1 and s2. The s2
services
  references information from the s1 services to satisfy its run-time properties.",
  "tenants": [],
  "approvals": [],
  "tested": false,
  "version": "1",
  "create-time": "2017-11-11T15:14:09.701Z",
  "created-by-user": "ibmuser",
  "last-modified-by-user": "ibmuser",
  "last-modified-time": "2017-11-11T15:22:17.595Z",
  "generated-name": "S1_S2.1.default",
  "domain-name": "default",
  "SAF-resources": [],
  "automatic-security": true,
  "published-timestamp": "2017-11-11T15:22:17.595Z",
  "archived-timestamp": "2017-11-11T15:25:08.396Z",
  "template-type": "composite",
  "composite-cluster": false,
  "composite-definition": [
    {
      "sequence": 1,
      "number-of-instances": 1,
      "connectors": [],
      "missing": true,
      "description": "",
      "prompt-variables": [],
      "published-template-name": "s1",
      "software-type": ""
    }
  ],
}

```



```

{
  "sequence": 2,
  "number-of-instances": 1,
  "connectors": [
    {
      "variable-name": "INS",
      "source-template": "s1",
      "source-variable-name": "registry-instance-Name",
      "not-valid": true
    },
    {
      "variable-name": "WELSHIE",
      "source-template": "s1",
      "source-variable-name": "WELSHIE",
      "not-valid": true
    }
  ],
  "missing": false,
  "description": "",
  "prompt-variables": [
    {
      "name": "CMD",
      "label": "CMD",
      "description": "CMD",
      "type": "string",
      "value": "S BCTEST",
      "required": false,
      "choices": null,
      "regex": ".{1,1000000}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "CMD",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "WELSHIE",
      "label": "name",
      "description": "This variable contains the name of a welsh springer
spaniel.",
      "type": "string",
      "value": "Scout",
      "required": false,
      "choices": null,
      "regex": ".*",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Name of a Welsh Springer Spaniel",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": ".*"
    },
    {
      "name": "INS",
      "label": "INS",
      "description": "INS",
      "type": "string",
      "value": "Instructions",
      "required": false,
      "choices": null,
      "regex": ".{1,1000000}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "INS",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    }
  ],
  "published-template-name": "s2",
  "software-type": "MIX"
},
{
  "composite-variable-input-file": "",
  "composite-variable-input-file-original-source": "",
  "composite-variable-input-file-original-timestamp": "",
  "security-wf-info": null,
  "admin-documentation-file-original-source": null,

```

```

    "admin-documentation": null,
    "admin-documentation-type": null,
    "consumer-documentation-file-original-source": null,
    "consumer-documentation": null,
    "consumer-documentation-type": null,
    "at-create-variables": [
        "s2.WELSHIE",
        "s2.UKEY",
        "s2.CMD",
        "s2.INS"
    ],
    "base-object-id": "f4feb4e9-f2e3-4121-a483-fa4bf10282b9",
    "admin-documentation-mime-type": null,
    "consumer-documentation-mime-type": null,
    "provisioning-version": "1400"
}

```

Get a software services template history

Use this operation to retrieve the history for a software services template in the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/history
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<object-id>

Identifies the template for which history is to be retrieved.

Query parameters

None.

Description

This operation retrieves the history for a software services template.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 245](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the template. Table 179 on page 245 lists the fields in the history object.

Table 179. Response from a get request: History object

Field	Type	Valid for Template Type	Description
action-type	String	Standard, Composite	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none">• Create• Add approval• Approve• Archive• Modify• Publish• Refresh• Reject• Remove approval• Run• Test run• Security complete• Update approval
user	String	Standard, Composite	The user who performed the action.
action-time	String	Standard, Composite	The time that the action was taken.
action-details	String	Standard, Composite	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

Example HTTP interaction

In Figure 78 on page 245, a request is submitted to retrieve the history for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/c06b4ba7-f72a-491e-8d63-5a38b4a4e4a3/history
```

Figure 78. Sample request to retrieve a software template history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:15.791Z",
      "action-details": "Created template"
    }
  ],
}
```

```
{
  "action-type": "Publish",
  "user": "ibmuser",
  "action-time": "2020-12-14T14:41:24.860Z",
  "action-details": "Published template"
}
```

Get software services template documentation

Use this operation to retrieve software services template documentation from the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/admin
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/consumer
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template to retrieve.

documentation/admin

Causes the administrator documentation file to be retrieved.

documentation/consumer

Causes the consumer documentation file to be retrieved.

Query parameters

None.

Description

This operation retrieves software services template documentation from the catalog.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in software services template documentation being retrieved.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

The documentation file in the associated mime type.

Example HTTP interaction

In Figure 79 on page 247, a request is submitted to retrieve the consumer documentation for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/5ccbad22-94fd-4b31-bb2b-95aa8602cc48/documentation/consumer
```

Figure 79. Sample request to retrieve software services template documentation

Get prompt variables for a software services template

Use this operation to retrieve the variables that are required to run the software services template and for which a prompt can be used to obtain the value.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/prompt-variables
```

In this request

<object-id>

Identifies the software services template to be retrieved.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation retrieves the variables for which a prompt can obtain the value.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 248](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the prompt variables. See [Table 180 on page 248](#), , and [Table 181 on page 248](#).

Table 180. Response from a get prompt variables request				
Field	Type	Required/ Optional	Valid for Template Type	Description
prompt-variables	Array of objects	Required	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See Table 181 on page 248 .
composite-prompt-variables	Array of objects	Required	Composite	Array of composite prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the composite software services template. See Table 182 on page 249 .

Table 181. Response from a get request: Prompt-Variable-Object		
Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.

Table 181. Response from a get request: Prompt-Variable-Object (continued)		
Field	Type	Description
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Fields of type String default to null.

Table 182. Response from a get request: Composite-Prompt-Variable-Object		
Field	Type	Description
published-template-name	String	The name of the published template in the composite template that the prompt-variables field is associated with.
prompt-variables	Array of objects	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the published-template-name software services template as part of the composite software services template. See Table 181 on page 248 .

Example HTTP interactions

Figure 80 on page 249 shows a request to retrieve the prompt variables for a standard template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b2846a/prompt-variables
```

Figure 80. Sample request to retrieve prompt variables

The following is the response body for the request.

```
{
  "prompt-variables": [
    {
      "name": "WELSHIE",
      "label": "name",
      "description": "This variable contains the name of a welsh springer spaniel.",
      "type": "string",
      "value": "Scout",
      "required": false,
      "choices": null,
      "regex": ".*",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Name of a Welsh Springer Spaniel",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": ".*"
    }
  ]
}
```

Figure 81. Response body for the GET prompt variables request

Figure 82 on page 250 shows a request to retrieve the prompt variables for a composite template.

Figure 82. Sample request to retrieve prompt variables, composite template

The following is the response body for the request.

```
{
  "composite-prompt-variables": [
    {
      "prompt-variables": [],
      "published-template-name": "s1"
    },
    {
      "prompt-variables": [
        {
          "name": "CMD",
          "label": "CMD",
          "description": "CMD",
          "type": "string",
          "value": "S BCTEST",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "CMD",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        },
        {
          "name": "WELSHIE",
          "label": "name",
          "description": "This variable contains the name of a welsh springer
spaniel.",
          "type": "string",
          "value": "Scout",
          "required": false,
          "choices": null,
          "regex": ".*",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "Name of a Welsh Springer Spaniel",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": ".*"
        },
        {
          "name": "INS",
          "label": "INS",
          "description": "INS",
          "type": "string",
          "value": "Instructions",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "INS",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        }
      ],
      "published-template-name": "s2"
    }
  ]
}
```

Figure 83. Response body for the GET prompt variables request, composite template

Get source information for a software services template

Use this operation to retrieve source information for a software services template .

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/sources
```

In this request

<object-id>

Identifies the software services template for which information is to be retrieved.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation retrieves source information for a template, which includes details of the original source paths that were provided, and whether the files have been changed since the last time the template was updated with the source paths.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 251](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object. See [Table 183 on page 251](#).

Table 183. Response from a get source request			
Field	Type		Description
action-definition-file	Source-Info-Object	Standard	Details for the action definition file. See Table 184 on page 252 .

Table 183. Response from a get source request (continued)			
Field	Type		Description
workflow-definition-file	Source-Info-Object	Standard	Details for the workflow definition file. See Table 184 on page 252 .
workflow-variable-input-file	Source-Info-Object	Standard	Details for the workflow variable input file. See Table 184 on page 252 .
composite-variable-input-file	Source-Info-Object	Composite	Details for the composite variable input file. See Table 184 on page 252 .

Table 184. Response from a get request: Source-Info-Object		
Field	Type	Description
original-source-path	String	The original source path provided for the file.
out-of-sync	boolean	Indicates if the file that is associated with the template matches the original source file. The value is false if the current file that is associated with the template matches the original source file, and true if the current file that is associated with the template differs from the original source file, or if the original source file is not found.

Example HTTP interactions

Figure 84 on page 252 shows a request to retrieve the source information for a standard template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463/sources
```

Figure 84. Sample request to retrieve source information

The following is the response body for the request.

```
{
  "action-definition-file": {
    "original-source-path": "/u/wfStandard/a.xml",
    "out-of-sync": false
  },
  "workflow-definition-file": {
    "original-source-path": "/u/wfStandard/p.xml",
    "out-of-sync": false
  },
  "workflow-variable-input-file": null
}
```

Figure 85. Response body for the get source request

Figure 86 on page 252 shows a request to retrieve the source information for a composite template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/asdfasdf-asdfasdf-asdfasdf-asdfas/sources
```

Figure 86. Sample request to retrieve source information for a composite template

The following is the response body for the request.

```
{
  "composite-variable-input-file": null
}
```

Figure 87. Response body for the get source request for a composite template

List the software services templates

You can use this operation to list the software services templates that are defined in the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software services template service. The following value is valid: `1.0`.

Query parameters

You can specify the following query parameters on this request. Objects matching all query parameters are returned.

domain-name

Optional, specifies the domain name.

name

Optional, regular expression, specifies the external name of the software services template.

owner

Optional, specifies the user ID or group ID that identifies the owner of the software services template.

software-type

Optional, specifies the type of software being provisioned.

state

Optional, regular expression, specifies the state.

template-type

Optional, specifies the type (standard or composite).

If you specify no query parameters, then all software services templates are returned.

Description

This operation lists the software services templates in the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a list of software services templates being retrieved. A response body is provided, as described in [“Response content” on page 254](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services templates. See [Table 185 on page 254](#), [Table 186 on page 254](#).

Table 185. Array of objects

Field	Type	Description
scc-list	Array of objects	Array of software services template objects. The array is filter based on any query parameters that were provided.

Table 186. Fields for each software services template

Field	Type	Value Returned for Template Type	Description
generated-name	String	Standard, Composite	The generated name for the software services template.
object-id	String	Standard, Composite	The unique ID that identifies the software services template.
base-object-id	String	Standard	The object ID that is associated with all of the versions of the software services template.
name	String	Standard, Composite	Descriptive name for the software services template.
version	String	Standard	Version of the software services template.
owner	String	Standard, Composite	User ID of the software services template owner.
state	String	Standard, Composite	Indicates the status of the software services template. See “State values” on page 256 .
description	String	Standard, Composite	Description of the software services template.
action-definition-file	String	Standard	Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template.
software-id	String	Standard	A short, arbitrary value that identifies the software that is being provisioned.
software-name	String	Standard	Name of the software that is being provisioned.
software-type	String	Standard	Identifies the type of software that is being provisioned.
software-version	String	Standard	Version of the software that is being provisioned.
workflow-definition-file	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow.

Table 186. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
workflow-id	String	Standard	Workflow ID. A short, arbitrary value that identifies the workflow.
workflow-vendor	String	Standard	Name of the vendor that provided the workflow definition file.
workflow-version	String	Standard	Version of the workflow definition file.
workflow-variable-input-file	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
domain-name	String	Standard, Composite	The name of the domain that the template resides in.
create-time	String	Standard, Composite	The time that this object was created, in ISO 8601 format.
created-by-user	String	Standard, Composite	The user that created this object.
last-modified-time	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.
last-modified-by-user	String	Standard, Composite	The user that last updated this object.
template-type	String	Standard, Composite	Identifies the type of template: standard Defines a single software service. composite Consists of multiple published templates that are provisioned together.
composite-cluster	boolean	Optional	Indicates whether child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
composite-definition	Array of objects	Composite	An array of objects that define the composite template (limited form). See Table 187 on page 257 .
composite-variable-input-file	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files. The variable names are in the following format: <standard-template-name>.<atcreate-variable> For example: CICS.startup=10

Table 186. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
provisioning-version	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Standard, Composite	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.

Note: In Table 186 on page 254, the **Value Returned for Template Type** column shows the type of template for which a value is returned. For other template types, null is returned.

State values

archived

The entry is hidden from consumers. You can make it available again with the **Publish** action.

corrupted

The contents of the software services template are missing or incorrect. Delete the template.

draft

The entry is in the edit state and visible only to the owner and the administrator. No approvals are required for this template. The entry can be tested with the Test a software services template API.

draft_approved

The software services template is in edit state and all the approvals that are associated with the template and the respective runAs user IDs have been received. The entry can be tested with the Test a software services template API.

draft_pending_approvals

The software services template is in edit state and one or more associated approvals has not been approved. The entry cannot be tested (with the Test a software services template API) in this state.

draft_missing_required_approver

One or more of the definition files contains a runAsUser element without a corresponding approver element. Either an approver element must be added for the runAsUser element, or a domain or general approver must be added for the software services template. The entry cannot be tested (with the Test a software services template API) in this state.

draft_rejected

The template is in a draft state and one or more approvers rejected an approval. The entry cannot be tested (with the Test a software services template API) in this state.

missing_required_member

One or more of the members of a composite template that was referenced in the composite definition is not available. This state applies only to published or archived composite templates.

pending_security_update

Permission to access the software services template is being processed. No API requests are allowed for the software services template until the security processing is complete.

published

The entry is locked and visible to consumers.

security_update_failed

Security access setup related to the software services template failed. Only the view and delete API requests are available.

Table 187. Contents of composite-definition object

Field	Type	Required/optional	Description
sequence	integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
missing	boolean	Required	<ul style="list-style-type: none"> • true if no published template is available that is related to the original version used when the template was defined • false if a published template exists that satisfies the published template requirement
description	String	Required	Description of the software services template.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
software-type	String	Required	Type of software that is being provisioned.

Example HTTP interaction

Figure 88 on page 257 shows a request to retrieve a list of software services template.

```
GET https://pev243.pok.ibm.com/zosmf/provisioning/rest/1.0/scc
```

Figure 88. Sample request to list software services templates

The following is a response body.

```

{
  "scc-list": [
    {
      "name": "s2",
      "version": "2",
      "owner": "ibmuser",
      "state": "published",
      "description": "",
      "generated-name": "s2.2.default",
      "object-id": "93d9c9dc-6b47-4222-89a6-f18764b28463",
      "base-object-id": "f9211dbd-a7e8-44ca-a5b3-a4eed0a21f69",
      "domain-name": "default",
      "action-definition-file": "definition/a.xml",
      "software-id": "prodID",
      "software-name": "Product Name",
      "software-type": "MIX",
      "software-version": "Version 1",
      "workflow-definition-file": "definition/p.xml",
      "workflow-id": "wfID",
      "workflow-vendor": "IBM",
      "workflow-version": "1.0",
      "workflow-variable-input-file": "",
      "create-time": "2017-11-11T15:13:10.909Z",
      "created-by-user": "ibmuser",
      "last-modified-by-user": "ibmuser",
      "last-modified-time": "2017-11-11T15:13:18.062Z",
      "template-type": "standard",
      "composite-parents": [
        "S1_S2"
      ],
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    },
    {
      "name": "S1_S2",
      "version": "1",
      "owner": "ibmuser",
      "state": "missing_required_member",
      "description": "This is a composite template that brings up services s1 and s2. The s2 services references information from the s1 services to satisfy its run-time properties.",
      "generated-name": "S1_S2.1.default",
      "object-id": "5f746d4c-ad24-4355-99d3-b83466ce4492",
      "base-object-id": "f4feb4e9-f2e3-4121-a483-fa4bf10282b9",
      "domain-name": "default",
      "create-time": "2017-11-11T15:14:09.701Z",
      "created-by-user": "ibmuser",
      "last-modified-by-user": "ibmuser",
      "last-modified-time": "2017-11-11T15:22:17.595Z",
      "template-type": "composite",
      "composite-cluster": false,
      "composite-definition": [
        {
          "sequence": 1,
          "number-of-instances": 1,
          "missing": true,
          "description": "",
          "published-template-name": "s1",
          "software-type": ""
        },
        {
          "sequence": 2,
          "number-of-instances": 1,
          "missing": false,
          "description": "",
          "published-template-name": "s2",
          "software-type": "MIX"
        }
      ],
      "composite-variable-input-file": "",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    }
  ],
}

```



```

{
  "name": "s1",
  "version": "2",
  "owner": "ibmuser",
  "state": "archived",
  "description": "",
  "generated-name": "s1.2.default",
  "object-id": "e214615b-ae4a-407c-8408-20b45b1a3472",
  "base-object-id": "08ead9fe-59f0-46c7-a1fb-8d5f9b39f08e",
  "domain-name": "default",
  "action-definition-file": "definition/a.xml",
  "software-id": "prodID",
  "software-name": "Product Name",
  "software-type": "MIX",
  "software-version": "Version 1",
  "workflow-definition-file": "definition/p.xml",
  "workflow-id": "wfID",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0",
  "workflow-variable-input-file": "",
  "create-time": "2017-11-11T15:09:59.963Z",
  "created-by-user": "ibmuser",
  "last-modified-by-user": "ibmuser",
  "last-modified-time": "2017-11-11T15:27:42.184Z",
  "template-type": "standard",
  "composite-parents": [],
  "provisioning-version": "1400",
  "provisioning-version-supported": true
}
]
,
}

```

Publish a software services template

You can use this operation to publish a software services template. The publish operation locks the template, preventing any further modification, and creates a public copy of it.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/publish
```

In this request

<object-id>

Identifies the software services template to be published.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation publishes a software services template.

Publishing a new version of a standard template automatically archives any composite templates that include it. The user can either republish an archived affected composite template or create a new version of it. The composite template can be published only if all connector information is supplied at the time of the publish operation.

On successful completion, HTTP status code 200 Ok is returned, indicating that the request resulted in a software services template being published.

The software services template must be in the draft, draft approved, or archived state.

To work with a published software services template, use the REST APIs that are described in [“Published software service template services” on page 283](#).

Request content

The request body is optional. It contains a JSON object that describes the publish operation. See [Table 188 on page 260](#).

Table 188. Request content for the software services template request			
Parameter	Type	Required or Optional	Description
archive-existing	boolean	Optional	<p>If set to true, indicates that if a published entry with this name already exists, that entry should be moved into the archived state, and publish this one instead.</p> <p>If set to false, indicates that if a published entry with this name already exists, the request should fail. False is the default if this parameter is not specified.</p> <p>If no published entry with this name already exists, then this flag is ignored.</p>
ignore-test	boolean	Optional	<p>If set to true, indicates a publish of the template does not require a test run to be performed.</p> <p>If set to false, a test run must be performed before a publish can be performed.</p> <p>If this parameter is not specified, then the value defaults to false.</p>
ignore-source-change	boolean	Optional	<p>If set to true, indicates that the publish of the entry is not restricted by the change in the original source that was used on the create or modify of the entry.</p> <p>If this parameter is not specified, then the value defaults to false.</p>

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 0k is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 189 on page 261](#) lists the fields in the JSON object.

Table 189. Response from the software services template request		
Field	Type	Description
composites-affected	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 190. Composite-affected object		
Field	Type	Description
name	String	Name of the composite template that was affected by an update to a standard template.
state	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

Example HTTP interaction

[Figure 89 on page 261](#) shows a request to publish a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/e214615b-ae4a-407c-8408-20b45b1a3472/actions/publish
```

Figure 89. Sample request to publish a software services template

The response body is as follows.

```
{
  "composites-affected": [
    {
      "name": "S1_S2",
      "state": "archived"
    }
  ]
}
```

Test a software services template

You can use this operation to test a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/test
```

In this request

<object-id>

Identifies the software services template to be published.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation lets you perform a test run of a software services template. The test run creates a workflow entry, starts the workflow entry, and creates a software services registry entry in being provisioned state.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the entry in the registry was created.

All approvals for a software services template must be approved before it can be tested. The software services template must be in the draft or draft approved state.

The software services template must be in the draft or draft_approved state.

Request content

The request content is expected to contain a JSON object that describes the test run. See [Request content for the software services template request](#) and [Table 192 on page 263](#).

Table 191. Request content for the test software services template request			
Field name	Type	Required or optional	Description
input-variables	Array of objects	Optional	Array of variable objects containing variables to be used for running the software services template. For a composite template, the variable naming convention is as follows: <published-template-name>.<atCreate-variable-name>. See Table 192 on page 263 .
user-data-id	String	Optional	ID of user-data. The user-data-id and user-data values are associated with the software services instance that is created and are returned with requests for the software services instance.
user-data	String	Optional	User-supplied data to be associated with the software services instance. Only allowed if user-data-id is also provided.
tenant-name	String	Optional	Required if the template is associated with more than one tenant
account-info	String	Optional	Account information to use in the JCL JOB statement. By default, it is the account information that is associated with the tenant resource pool
systems-nicknames	Array of Strings	Optional	Each string is the nickname of the system upon which to provision the software service defined by the template. The field is required if the resource pool associated with the tenant used for this operation is not set up to automatically select a system. Only one nickname is allowed. If the field is provided it is validated.

Table 191. Request content for the test software services template request (continued)

Field name	Type	Required or optional	Description
expiration-period	Integer	Optional	Number of days the instance will be kept provisioned after it is successfully provisioned. A value of 0 indicates that the instance does not expire. If not specified, this value defaults to the <code>rdp-instance-expiration-limit</code> value for the template.

Table 192. Runtime properties

Field	Type	Description
name	String	Name of the runtime property.
value	String	Value of the runtime property.
sequence	String	Provisioning sequence of the runtime property.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template request. [Table 191 on page 262](#) lists the fields in the JSON object.

Table 193. Response from a test software services template request

Field	Type	Description
registry-info	Object	Object mapping matching the return response body from a registry create. For a composite type template, this field reflects the parent registry instance response body.
workflow-info	Object	Object mapping matching the return response body from a workflow create. This field is not returned for a composite template.
composite-children-registry-info	Array of objects	For a composite type template, this field contains an array of composite child registry information objects. See Table 194 on page 264 .
system-nickname	String	Nickname of the system that the service is provisioned on.

Table 194. Composite child registry information objects		
Field	Type	Description
sequence	Integer	The order in which the child registry instances are being provisioned, starting with 1. The deprovisioning order is the reverse.
object-name	String	The name of the newly created object.
object-id	String	The ID of the newly created object. This object ID is to be used on further requests to the object.
object-uri	String	The URI of the newly created object
external-name	String	The external name of the newly created object.

Example HTTP interactions

Figure 90 on page 264 shows a request to test a standard software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/6e39d092-745a-4e81-8b7b-d3e1327ad230/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "CMD", "value": "S BCTEST" },
    { "name": "INS", "value": "Instructions" },
    { "name": "WELSHIE", "value": "Truepenny Traveling a Trail of Hope" },
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 90. Sample request to test a standard software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "MIX_2",
    "object-id": "c35de2ea-3d6c-47ec-bc32-62b9013ffcd5",
    "object-uri": "/zosmf/provisioning/rest/1.0/scc/c35de2ea-3d6c-47ec-bc32-62b9013ffcd5",
    "external-name": "MIX_SCOUT01",
    "system-nickname": "SY1"
  },
  "workflow-info": {
    "workflowKey": "540ef4fa-754b-40dd-9951-0e80edd1ec3b",
    "workflowDescription": "Mix1 workflow",
    "workflowID": "wfID",
    "workflowVersion": "1.0",
    "vendor": "IBM"
  },
  "system-nickname": "SY1"
}
```

Figure 91. Sample response body

Figure 92 on page 264 shows a request to test a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "s1.CMD", "value": "S BCTEST" },
    { "name": "s1.WELSHIE", "value": "Tucker" }
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 92. Sample request to test a composite software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "SCOUT_3",
    "object-id": "6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "external-name": "SCOUT_SCOUT00",
    "system-nickname": "SY1"
  },
  "system-nickname": "SY1",
  "composite-children-registry-info": [
    {
      "sequence": 1,
      "object-name": "MIX_5",
      "object-id": "725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "external-name": "MIX_SCOUT01"
    },
    {
      "sequence": 2,
      "object-name": "MIX_6",
      "object-id": "b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "external-name": "MIX_SCOUT02"
    }
  ]
}
```

Figure 93. Sample response body

Refresh a software services template

You can use this operation to refresh the files that are associated with a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/refresh
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.

The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

<approval-object-id>

Identifies the approval object to delete.

Query parameters

None.

Description

This operation obtains the latest contents of the files that are associated with a software services template. For a standard template, this includes the workflow definition XML file, the actions XML file, the workflow variable input file if one is specified, and any documentation files that are provided. For a composite template, this includes the composite variable input file if one is specified, and any documentation files that are provided. The information in the software services template is updated to reflect the latest contents, including timestamps, of those files. The files are located by the original source paths.

Refresh causes all approvals to be reset.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

Example HTTP interaction

In [Figure 94 on page 266](#), a request is submitted to refresh a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/d0166782-4e18-4b07-a075-c8946c88e068/actions/refresh
```

Figure 94. Sample request to refresh a software services template

Archive a software services template

You can use this operation to archive a published software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/archive
```

In this request

<object-id>

Identifies the software services template to be archived.

<version>

Is the URI path variable `<version>` that identifies the version of the z/OSMF software services template service. The following value is valid: `1.0`.

Query parameters

None.

Description

This operation lets you archive a software services template. This puts the software services template in an archived state.

The software services template must be in a published state.

On successful completion, HTTP status code 200 Ok is returned, indicating that the archive was successful.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 OK is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 167 on page 228](#) lists the fields in the JSON object.

Table 195. Response from the software services template request		
Field	Type	Description
composites-affected	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 196. Composite-affected object		
Field	Type	Description
name	String	Name of the composite template that was affected by an update to a standard template.
state	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

Example HTTP interaction

In [Figure 95 on page 267](#), a request is submitted to archive a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/e214615b-ae4a-407c-8408-20b45b1a3472/actions/archive
```

Figure 95. Sample request to archive a software services template

The following is the response body for the request.

The response body is as follows.

```
{
  "composites-affected": [
    {
      "name": "S1_S2",
      "state": "missing_required_member"
    }
  ]
}
```

```
} ]
```

Add an approval for a software services template

You can use this operation to create an approval record for a software services template. The approval record associates a user ID with the software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
```

In this request

<object-id>

Identifies the software services template that the approval is associated with.

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation lets you create a new general approval record for a software services template. It returns a unique approval object ID that identifies the approval. The approval is for all of the contents of the software services template and is not associated with a specific step or action.

The software services template must be in one of the draft states.

All approvals for a software services template must be approved before it can be published or tested. Once all the approvals are approved the state of the entry is updated to draft_approved.

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the approval was created.

Request content

The request content is expected to contain a JSON object that describes the approval record. See [Request content for the software services template request](#).

Table 197. Request content for the add approval request			
Field name	Type	Required or optional	Description
user-id	String	Required	User ID associated with this approval.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 201 (Normal) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template request. [Table 197 on page 268](#) lists the fields in the JSON object.

Table 198. Response from an add approval request		
Field	Type	Description
object-id	String	Object ID of the newly created approval. The object ID is to be used in subsequent requests to the session.
object-uri	String	URI of the newly created approval.

Example HTTP interaction

In [Figure 96 on page 269](#), a request is submitted to add an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals
{
  "user-id": "nick"
}
```

Figure 96. Sample request to add an approval record for a software services template

The following is the response body for the request.

```
{
  "object-id": "eeb4f5a3-d883-4190-9961-412306707426",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/eeb4f5a3-d883-4190-9961-412306707426"
}
```

Figure 97. Sample response body

Get an approval for a software services template

You can use this operation to retrieve an approval for a software services template.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
    <approval-object-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with..

<approval-object-id>

Identifies the approval to retrieve.

Query parameters

None.

Description

This operation retrieves an approval for a software services template.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in an approval being retrieved. A response body is provided, as described in [“Response content” on page 270](#)

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the approval. [Table 199 on page 270](#) lists the fields in the JSON object.

Table 199. Response from a get approval request		
Field	Type	Description
status	String	Status of the approval for this object: pending, approved, or rejected.
comment	String	Comment that is associated with the change in status from pending to either approved or rejected.
description	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition.
approvers	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
status-update-by	String	User ID that performed the last approve or reject action for this approval object.
time-of-update	String	The last time this object was updated, in ISO 8601 format.
run-as-user	String	The runAsUser user ID that the approval object is for. This applies only to action_definition and step_definition types.

Table 199. Response from a get approval request (continued)		
Field	Type	Description
type	String	Type of approval object: general, domain, action_definition, or step_definition.
object-id	String	Unique object ID representing this approval object.
workflow-file	String	Workflow file definition that is associated with this runAsUser user ID.
variable-input-file	String	Specifies the variable input file that is associated with this runAsUser user ID.
step-name	String	Workflow file definition step that is associated with this runAsUser user ID.
called-by-step-name	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
called-by-workflow-file	String	Workflow definition file that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
actions-file	String	Actions definition file that is associated with this runAsUser user ID.
action-name	String	Action defined in the actions definition file that is associated with this runAsUser user ID.

Example HTTP interaction

In Figure 98 on page 271, a request is submitted to get an approval record for a software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/
3f8ca645-f872-42b6-b0fc-3c6a9e470fcc/approvals/11aeb028-9a0a-45eb-a005-4a9460126c3a
```

Figure 98. Sample request to get an approval record for a software services template

The following is the response body for the request.

```
{
  "status": "missing_approver",
  "comment": null,
  "description": "The approver element originates from the 'Auto-Step' step in the /u/wfSuspend/p.xml
which is the primary workflow definition file.",
  "type": "step_definition",
  "object-id": "11aeb028-9a0a-45eb-a005-4a9460126c3a",
  "user-ids": [],
  "status-update-by": null,
  "time-of-update": null,
  "run-as-user": "${instance-rau}",
  "workflow-file": "/u/wfSuspend/p.xml",
  "variable-input-file": null,
  "step-name": "Auto-Step",
  "called-by-step-name": null,
  "called-by-workflow-file": null,
  "actions-file": null,
  "action-name": null,
  "run-as-user-dynamic": true
}
```

Figure 99. Sample response body

List the approvals for a software services template

You can use this operation to list all of the approvals for a software services template.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

Query parameters

None.

Description

This operation retrieves all of the approval for a software services template.

On successful completion, HTTP status code 200 (Normal) is returned. A response body is provided, as described in [“Response content” on page 272](#)

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned.

Response content

On successful completion, the service returns a JSON object. See [Table 200 on page 272](#) and [Table 201 on page 273](#).

Table 200. Response from a list approvals request		
Field	Type	Description
approvals	Array of objects	Array of Approval-Object containing information about the approvals associated with this software services template.

Table 201. Response from a get approval request

Field	Type	Description
status	String	Status of the approval for this object: pending, approved, or rejected.
comment	String	Comment that is associated with the change in status from pending to either approved or rejected.
description	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition.
approvers	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
status-update-by	String	User ID that performed the last approve or reject action for this approval object.
time-of-update	String	The last time this object was updated, in ISO 8601 format.
run-as-user	String	The runAsUser user ID that the approval object is for. This applies only to action_definition and step_definition types.
type	String	Type of approval object: general, domain, action_definition, or step_definition.
object-id	String	Unique object ID representing this approval object.
workflow-file	String	Workflow file definition that is associated with this runAsUser user ID.
variable-input-file	String	Specifies the variable input file that is associated with this runAsUser user ID.
step-name	String	Workflow file definition step that is associated with this runAsUser user ID.
called-by-step-name	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
called-by-workflow-file	String	Workflow definition file that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
actions-file	String	Actions definition file that is associated with this runAsUser user ID.
action-name	String	Action defined in the actions definition file that is associated with this runAsUser user ID.

Example HTTP interaction

In Figure 100 on page 273, a request is submitted to get an approval record for a software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/3f8ca645-f872-42b6-b0fc-3c6a9e470fcc/approvals
```

Figure 100. Sample request to list the approval records for a software services template

The following is the response body for the request.

```
{
  "approvals": [
    {
      "status": "missing_approver",
      "comment": null,
      "description": "The approver element originates from the 'Auto-Step' step in the /u/wfSuspend/p.xml
        which is the primary workflow definition file.",
      "type": "step_definition",
      "object-id": "11aeb028-9a0a-45eb-a005-4a9460126c3a",
      "approvers": [],
      "status-update-by": null,
      "time-of-update": null,
      "run-as-user": "${instance-rau}",
      "workflow-file": "/u/wfSuspend/p.xml",
      "variable-input-file": null,
      "step-name": "Auto-Step",
      "called-by-step-name": null,
      "called-by-workflow-file": null,
      "actions-file": null,
      "action-name": null,
      "run-as-user-dynamic": true
    },
    {
      "status": "pending",
      "comment": null,
      "description": "The approver element originates from the primary action file.",
      "type": "action_definition",
      "object-id": "263c87d7-4043-4fe4-895f-ccd3ad092966",
      "approvers": [
        "zosmfad",
        "ibmuser",
        "agrp1",
      ],
      "status-update-by": null,
      "time-of-update": null,
      "run-as-user": "zosmfad",
      "workflow-file": null,
      "variable-input-file": null,
      "step-name": null,
      "called-by-step-name": null,
      "called-by-workflow-file": null,
      "actions-file": "/u/wfSuspend/a.xml",
      "action-name": "command1",
      "run-as-user-dynamic": false
    }
  ]
}
```

Figure 101. Sample response body

Approve an approval record for a software services template

You can use this operation to approve the contents of approval record for a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/  
<approval-object-id>/actions/approve
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

<approval-object-id>

Identifies the approval object to approve.

Query parameters

None.

Description

This operation approves the contents of an approval record for the software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

Request content

The request content is expected to contain a JSON object that describes the approval. See [Table 202 on page 275](#).

Table 202. Request content for the software services template request			
Field name	Type	Required or optional	Description
comment	String	Optional	Text describing the approval.

Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

Example HTTP interaction

In [Figure 102 on page 276](#), a request is submitted to approve an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/dacea656-ffbe-48ce-a193-575161ff9d43/actions/approve
```

Figure 102. Sample request to approve an approval record for a software services template

Batch approve approval records for a software services template

You can use this operation to batch approve or reject the contents of one or more approval records for a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
/actions/update
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approvals are associated with.

Query parameters

None.

Description

This operation performs batch approval of the contents of one or more approval records for the software services template.

On successful completion, HTTP status code 204 (No content) is returned.

The software services template must be in one of the draft states.

Request content

The request content is expected to contain a JSON object that describes the approval. See [Table 203 on page 276](#).

Table 203. Request content for the software services template request			
Field name	Type	Required or optional	Description
approve	Array of approval objects	Optional	Identify what is being approved. See Table 204 on page 277 .
reject	Array of approval objects	Optional	Identify what is being rejected. See Table 204 on page 277 .

Table 204. Approval objects

Field name	Type	Required or optional	Description
approval-object-ids	Array of Strings	Required	Strings that identify the approval objects on which to perform the specified approval or rejection.
comment	String	Optional	Text describing the approval.

Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (No Content) is returned.

Example HTTP interactions

In [Figure 103 on page 277](#), a request is submitted to approve an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "approve": {
    "approval-object-ids": ["6b5e7b0d-34d0-4ca8-8b73-4c9f6d178ad2"],
    "comment": "This is a comment"
  }
}
```

Figure 103. Sample request to perform batch approval for a software services template

In [Figure 104 on page 277](#), a request is submitted to reject approval records for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "reject": {
    "approval-object-ids": ["cc9ecc32-c32d-48d0-8d55-8349269f51ee",
                           "9e00c325-9bc3-47bb-b106-c7c41772eea3",
                           "aaeec169-6637-4e37-9f71-01838f1f1ce8"]
  }
}
```

Figure 104. Sample request to perform batch rejection of approvals for a software services template

In [Figure 105 on page 278](#), a request is submitted to approve and reject approval records for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "approve": {
    "approval-object-ids": ["6b5e7b0d-34d0-4ca8-8b73-4c9f6d178ad2"],
    "comment": "This is a comment"
  },
  "reject": {
    "approval-object-ids": ["cc9ecc32-c32d-48d0-8d55-8349269f51ee",
                           "9e00c325-9bc3-47bb-b106-c7c41772eea3",
                           "aaeec169-6637-4e37-9f71-01838f1f1ce8"]
  }
}
```

Figure 105. Sample request to perform batch approval and rejection of approvals for a software services template

Reject the use of a user ID with a software services template

You can use this operation to reject the use of your user ID with a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
<approval-object-id>/actions/reject
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

<approval-object-id>

Identifies the approval object to reject.

Query parameters

None.

Description

This operation rejects an approval that is associated with a software services template. Rejecting the approval means that your user ID is not allowed to be used with the software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

Request content

The request content is expected to contain a JSON object that describes the rejection. See [Table 205 on page 278](#).

Table 205. Request content for the software services template request			
Field name	Type	Required or optional	Description
comment	String	Optional	Text describing the rejection.

Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

Example HTTP interaction

In [Figure 106 on page 279](#), a request is submitted to reject an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/
dacea656-ffbe-48ce-a193-575161ff9d43/actions/reject

{
  "comment": "disagree with this, rework required"
}
```

Figure 106. Sample request to reject an approval record for a software services template

Delete an approval for a software services template

You can use this operation to delete an approval that is associated with a software services template.

HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
<approval-object-id>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

<approval-object-id>

Identifies the approval object to delete.

Query parameters

None.

Description

This operation deletes an approval that is associated with a software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

Example HTTP interaction

In [Figure 107 on page 280](#), a request is submitted to delete an approval record for a software services template.

```
DELETE https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/  
dacea656-ffbe-48ce-a193-575161ff9d43
```

Figure 107. Sample request to delete an approval record for a software services template

Set security complete for a software services template

You can use this operation to indicate that the required security setup has been completed for a software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/  
security_complete
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<object-id>

Identifies the software services template that the approval is associated with.

Query parameters

None.

Description

This operation indicates that the required security setup has been completed for a software services template. The template can move to the next state.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in the pending security update or pending security failed state.

Request content

None.

Authorization requirements

The user's z/OS user ID must be defined as a provisioning administrator or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

Example HTTP interaction

In [Figure 108 on page 281](#), a request is submitted to indicate that security setup is complete for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/d0166782-4e18-4b07-a075-c8946c88e068/  
actions/security_complete
```

Figure 108. Sample request to indicate security is complete for a software services template

Published software service template services

The published software service template services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services templates that are in the published state.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 46.

The published software services catalog contains a list of the software services templates that are in the published state.

Table 206 on page 283 lists the operations that the published software service template services provide.

Table 206. z/OSMF published software service template services: operations summary

Operation name	HTTP method and URI path
“Run a published software service template” on page 285	POST /zosmf/provisioning/rest/<version>/psc/<name>/actions/run
“Get a published software service template” on page 290	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
“Get a published software service template history” on page 302	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/history
“Get consumer documentation for a published software service template” on page 304	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/documentation/consumer
“Get prompt variables for a published software service template” on page 306	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/prompt-variables
“List the published software service templates” on page 311	GET /zosmf/provisioning/rest/<version>/psc/
“Modify a published software service template” on page 315	POST /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>

Authorization requirements

Use of the published software service template services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

The specific requirements for each published software service template service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 49](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 207. Response from a software services template request failure		
Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

Error logging

Errors from the software services template services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request succeeded and resulted in the creation of an object.

HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

HTTP 204 No content

The request succeeded, but no content is available to be returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 404 Not found

The requested resource does not exist.

HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

Related information

The run operation for a published template creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry. To work with a software services instance, use the REST APIs described in [“Software services instance services” on page 319](#).

Run a published software service template

Use this operation to run a software services template that is in the published state.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/psc/<name>/actions/run
```

In this request:

<version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<name>

Identifies the software services template to be run.

Query parameters

None.

Description

This operation creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry, with a state of being-provisioned.

To work with a software services instance, use the REST APIs described in [“Software services instance services” on page 319](#).

Request content

The request content is expected to contain a JSON object as described in [Table 208 on page 285](#) and [Table 209 on page 286](#).

Table 208. Request content for the run software services template request			
Field name	Type	Required or optional	Description
input-variables	Array of Objects	Optional	An array of required runtime property objects. See Table 209 on page 286 For a composite template, the variable naming convention is as follows: <published-template-name>.<atCreate-variable-name>.
domain-name	String	Optional	Required if the user has consumer authorization to more than one domain with this template name.
tenant-name	String	Optional	Required if the user has consumer authorization to more than one tenant in the same domain that contains this template name.
user-data-id	String	Optional	ID for the user data specified with user-data. Passed into the software services registry.

Table 208. Request content for the run software services template request (continued)

Field name	Type	Required or optional	Description
account-info	String	Optional	Account information to use in the JCL JOB statement. The default is the account information that is associated with the resource pool for the tenant.
user-data	String	Optional	User data that is passed into the software services registry. Can be specified only if user-data-id is provided.
systems-nicknames	Array of Strings	Optional	Each string is the nickname of the system upon which to provision the software service defined by the template. The field is required if the resource pool associated with the tenant used for this operation is not set up to automatically select a system. Only one nickname is allowed. If the field is provided it is validated.
expiration-period	Integer	Optional	Number of days the instance will be kept provisioned after it is successfully provisioned. A value of 0 indicates that the instance does not expire. If not specified, this value defaults to the <code>rdp-instance-expiration-limit</code> value for the template.

Table 209. Runtime properties

Field	Type	Description
name	String	Name of the runtime property.
value	String	Value of the runtime property.
sequence	String	Provisioning sequence of the runtime property.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 286](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object, as described in [Table 210 on page 287](#).

Table 210. Response from a run software services template request

Field	Type	Description
registry-info	Object	Object mapping that matches the response body returned from a registry create action. For a composite type template, this field reflects the parent registry instance response body.
workflow-info	Object	Object mapping that matches the response body returned from a workflow create action.
composite-children-registry-info	Array of objects	For a composite type template, this field contains an array of composite child registry information objects. See Table 211 on page 287 .
system-nickname	String	Nickname of the system that the service is provisioned on.

Table 211. Composite child registry information objects

Field	Type	Description
sequence	Integer	The order in which the child registry instances are being provisioned, starting with 1. The deprovisioning order is the reverse.
object-name	String	The name of the newly created object.
object-id	String	The ID of the newly created object. This object ID is to be used on further requests to the object.
object-uri	String	The URI of the newly created object
external-name	String	The external name of the newly created object.

Example HTTP interaction

In [Figure 109 on page 287](#), a request is submitted to run the software services template named bringUpDB2.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2/actions/run

{
  "input-variables": [{ "name": "CSQ_MQ_SSID", "value": "ZCT1" },
    { "name": "CSQ_CMD_PFX", "value": "ZCT1" }, { "name": "CSQ_ENVIRONMENT", "value": "TEST" } ],
  "domain-name": "default",
  "tenant-name": "tenant1"
}
```

Figure 109. Sample request to run a software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "QMGR_7",
    "object-id": "c5a8ecdd-db35-466b-aad9-cba0f33bb84b",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/c5a8ecdd-db35-466b-aad9-cba0f33bb84b"
  },
  "workflow-info": {
    "workflowKey": "ff96459f-27fa-490a-a3e4-4086649c12f3",
    "workflowDescription": "Procedure to provision a MQ for zOS Queue Manager",
    "workflowID": "ProvisionQueueManager",
    "workflowVersion": "1.0.1",
    "vendor": "IBM",
  },
  "system-nickname": "DUMBNODE"
}
```

Figure 110. Sample response body

Figure 111 on page 288 shows a request to test a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "s1.CMD", "value": "S BCTEST" },
    { "name": "s1.WELSHIE", "value": "Tucker" }
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 111. Sample request to test a composite software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "SCOUT_3",
    "object-id": "6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "external-name": "SCOUT_SCOUT00",
    "system-nickname": "SY1"
  },
  "system-nickname": "SY1",
  "composite-children-registry-info": [
    {
      "sequence": 1,
      "object-name": "MIX_5",
      "object-id": "725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "external-name": "MIX_SCOUT01"
    },
    {
      "sequence": 2,
      "object-name": "MIX_6",
      "object-id": "b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "external-name": "MIX_SCOUT02"
    }
  ]
}
```

Figure 112. Sample response body

Get a published software service template

Use this operation to retrieve a published software service template from the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<existing-entry-name>

Identifies the software services template to be retrieved.

Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

domain-name

Optional, string, specifies the domain name.

If you specify no query parameters, then all templates are returned.

Description

This operation retrieves a published software service template from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content”](#) on page 290.

Request content

None.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

See [“Authorization requirements”](#) on page 49.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. See [Table 213 on page 295](#), [Table 214 on page 296](#), [Table 215 on page 297](#), and [Table 216 on page 297](#).

Table 212. Response from a get software services template request

Field	Type	Valid for Template Type	Description
template-type	String	Standard	Identifies the type of template: standard Defines a single software service. composite Consists of multiple published templates that are provisioned together.
composite-definition	Array of objects	Composite	An array of objects that define the composite template. See Table 218 on page 298 .
composite-type	String	Standard	Type of composite template (software service).
composite-parents	Array of strings	Standard	An array of strings. Each string is a composite template that includes this standard template. For example: [c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]
composite-cluster	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
base-object-id	String	Standard	The object ID that is associated with all of the versions of the software services template.
generated-name	String	Standard, Composite	Generated name for the software services template.
object-id	String	Standard, Composite	Identifier for the software services template.
name	String	Standard, Composite	The name associated with the software services template.
version	String	Standard	Version of the software services template.
owner	String	Standard, Composite	User ID of the software services template owner.
state	String	Standard, Composite	Indicates the status of the software services template. It is always published. The entry is locked and visible to consumers.
description	String	Standard, Composite	Description of the software services template.
tenants	Array of Strings	Standard, Composite	Each string represents a tenant that the template is associated with.
domain-shared-tenants	Array of Strings	Standard, Composite	Each string represents a tenant in the domain that the template is associated with through the domain shared resource pool.

Table 212. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
domain-name	String	Standard, Composite	The domain the template is associated with.
approvals	Array of objects	Standard, Composite	Array of Approval-Object containing information about the approvals associated with this software services template. See Table 214 on page 296 .
action-definition-file	String	Standard	Location of the action definition file.
action-definition-file-original-source	String	Standard	Original user specified location of the action definition file
action-definition-file-original-timestamp	String	Standard	Last-modified time stamp for when the original action definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
actions	Array of objects	Standard	Array of Action-Object containing information about the actions associated with the template. See Table 215 on page 297 .
software-id	String	Standard	A short, arbitrary, value that identifies the software that is being provisioned.
software-name	String	Standard	Name of the software that is being provisioned.
software-type	String	Standard	Type of software that is being provisioned.
software-version	String	Standard	Version of the software that is being provisioned.
workflow-definition-file	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow
workflow-definition-file-original-source	String	Standard	Original user-specified location of the workflow definition file.
workflow-definition-file-original-timestamp	String	Standard	The last-modified time stamp for when the original workflow definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-id	String	Standard	A short, arbitrary value that identifies the workflow.
workflow-vendor	String	Standard	Name of the vendor that provided the workflow definition file.
workflow-version	String	Standard	Version of the workflow definition file.

Table 212. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
composite-variable-input-file	String	Composite	Location of the user-specified properties file that supplies values for one or more of the atCreate variables that are defined in the member standard template workflow definition files. The variable names are in the following format: <standard-template-name>.<atcreate-variable> For example: CICS.startup=10
composite-variable-input-file-original-source	String	Composite	Location of the composite variable input file, an optional properties file used to specify in advance the values for one or more of the atCreate variables.
composite-variable-input-file-original-timestamp	String	Composite	The last-modified time stamp for when the original composite variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-variable-input-file	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
workflow-variable-input-file-original-source	String	Standard	The original user-specified location of the workflow variable input file.
workflow-variable-input-file-original-timestamp	String	Standard	The last-modified time stamp for when the original variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
workflow-clean-after-provisioned	Boolean	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.
prompt-variables	Array of objects	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See Table 213 on page 295 .
public-variables	Array of strings	Standard	Array of strings that name the public variables for the template.

Table 212. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
at-create-variables	Array of strings	Standard	<p>Array of strings that name the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both.</p> <p>For a composite type template, any atCreate variable that was designated as a connector variable is excluded from the list. The variables are prefixed by the standard template name, for example MQ.BRING_UP.</p>
consumer-documentation-file	String	Standard, Composite	Location of the original file that provides information for consumers about the template.
consumer-documentation-type	String	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
admin-documentation-file	String	Standard, Composite	Location of a file that provides information for administrators about the template.
admin-documentation-type	String	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.
create-time	String	Standard, Composite	Time that this object was created, in ISO 8601 format.
create-by-user	String	Standard, Composite	User who created this object.
last-modified-time	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.
last-modified-by-user	String	Standard, Composite	User who last updated this object.
workflows-disposition	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.
jobs-disposition	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.
instances-disposition	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.
automatic-security	String	Standard	<p>Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization:</p> <ul style="list-style-type: none"> • true if the domain that the template is associated with is set up for automatic authorization • false if the domain that the template is associated with is set up for manual authorization.

Table 212. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
SAF-resources	Array of objects	Standard	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 217 on page 298 .
runAsUsers	Array of objects	Standard	Array of RunAsUser objects containing information about runAsUser IDs that are referenced by this template. See Table 220 on page 300 .
provisioning-version	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.

Table 213. Response from a get request: Prompt-Variable Object

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Fields of type String default to null.

<i>Table 214. Response from a get request: Approval-Object</i>		
Field	Type	Description
status	String	Status of the approval for this object: pending, approved, or rejected.
comment	String	Comment associated with the change in status from pending to either approved or rejected.
description	String	Additional detail provided if the approval is for a workflow definition that's associated with the action definition. For example: This workflow definition is associated with the <action-name> action.
user-ids	Array of strings	Each string in the array is a user ID. Any one of the user IDs in the array can approve or reject the item. The action of the last ID takes precedence.
status-update-by	String	The user ID that performed the last approve or reject action for this approval object
time-of-update	String	The last time this object was updated, in ISO 8601 format.
run-as-user	String	The runAsUser user ID that the approval object is for. This applies only when the type is action_definition or step_definition.
type	String	Type of approval: general (for the template), domain, action_definition, step_definition
object-id	String	Unique object id representing this approval object.
workflow-file	String	Workflow file definition associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
variable-input-file	String	Variable input file associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
step-name	String	Workflow file definition step associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
called-by-step-name	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
called-by-workflow-file	String	Workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.

Table 214. Response from a get request: Approval-Object (continued)

Field	Type	Description
actions-file	String	Actions file definition associated with this runAsUser user ID. Null if the user ID is not associated with an action or is a general approval.
action-name	String	Action defined in the actions file associated with this runAsUser user ID. Null if the user ID is not associated with an action or is a general approval.
run-as-user-dynamic	boolean	Indicates if the run-as-user ID value can change: true The run-as-user ID value is not final and can change during the processing of the workflow false The run-as-user ID is final and cannot change during the processing of the workflow.

Table 215. Response from a get request: Action-Object

Field	Type
name	String
type	String
command	String
workflow-definition-file	String
workflow-variable-input-file	String
workflow-variables	Variable[]
instructions	String
prompt-variables	The prompt variable objects that are associated with the action.
command-run-as-user-dynamic	boolean. Indicates if the command-run-as-user ID value can change: true The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content. false The command-run-as-user ID is final and cannot change during the processing of the command.

Table 216. Response from a get request: Variable-Object

Field	Type
name	String
value	String
visibility: public or private	String

Table 217. Response from a create request: SAF-resource object

Field	Type	Description
description	String	Description of the resource.
ids	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
groups	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
role	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
resource-class	String	The class associated with SAF resource.
resource-name	String	The name associated with the SAF resource.
required-access	String	The access required for the IDs and/or groups to be authorized successfully.
other-required-ids	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
audit-requirements	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Table 218. Response from a get request: composite-definition object

Field	Type	Description
sequence	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Indicates the number of child instances to be created using the template in a composite cluster.
published-template-name	String	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of Objects	<p>An array of connector object.</p> <p>The connector variables specified here take precedence over the variables field in this object and any variables specified in the workflow variable input file associated with the published template.</p> <p>See Table 219 on page 299.</p>

Table 218. Response from a get request: composite-definition object (continued)

Field	Type	Description
prompt-variables	Array of objects	<p>Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template.</p> <p>If specified, this overrides the array of prompt variables that are associated with the template specified with published-template-name. Only prompt variables that are already specified for the published-template-name can be specified. An empty array will translate into not prompting for any variables. If this field is not provided or set to null, then the prompt variables that are associated with published-template-name are used.</p> <p>If the connector variable-name is also a prompt-variable, then the connector takes precedence, and the variable is not promptable.</p>

Table 219. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.
not-valid	boolean	Required	<p>Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is:</p> <ul style="list-style-type: none"> • false, if all of the information is accurate • true, if one or more of the values are incorrect.

Table 220. Response from a get request: RunAsUser object

Field	Type	Description
description	String	Additional detail provided if the run-as-user is for a workflow definition that is associated with the action definition. Example: This workflow definition is associated with the <action-name> action.
approver-user-ids	Array of Strings	Array of strings where each string is a user ID that originates from the approver element that is associated with the runAsUser for the template step or action.
run-as-user	String	The runAsUser user ID that the approval object is for. This is applicable only to action_definition and step_definition type.
type	String	One of the following: action_definition or step_definition
workflow-file	String	The workflow file definition that is associated with this runAsUser user ID.
variable-input-file	String	The variable input file that is associated with this runAsUser user ID.
step-name	String	The workflow file definition step that is associated with this runAsUser user ID.
called-by-step-name	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the step in the parent workflow definition that called the workflow definition file that generated the approval object.
called-by-workflow-file	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the workflow definition that called the workflow definition file that generated the approval object.
actions-file	String	The actions file definition that is associated with this runAsUser user ID.
action-name	String	The action defined in the actions file that is associated with this runAsUser user ID.
run-as-user-dynamic	boolean	Indicates if the run-as-user ID value can change: true The run-as-user ID value is not final and can change during the processing of the workflow false The run-as-user ID is final and cannot change during the processing of the workflow.

Example HTTP interaction

In Figure 113 on page 300, a request is submitted to retrieve a standard software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2
```

Figure 113. Sample request to get a standard software services template

The following is the response body for the example GET request.

```
{
  "name": "mqCBA",
  "version": "1",
  "owner": "domadmin",
  "state": "published",
  "description": "This workflow provisions an MQ for z/OS Queue Manager",
  "tenants": [...],
  "actions": [...],
  "approvals": [],
  "tested": false,
  "generated-name": "mqCBA.1.default",
  "domain-name": "default",
  "action-definition-file": "definition/qmgrActions.xml",
  "action-definition-file-original-source": "/users/gg/mqCBA/definition/qmgrActions.xml",
  "action-definition-file-original-timestamp": "2016-11-18T20:00:42Z",
  "software-id": "5655-W97",
  "software-name": "IBM MQ for z/OS",
  "software-type": "QMGr",
  "software-version": "V8.0.0",
  "workflow-definition-file": "definition/provision.xml",
  "workflow-definition-file-original-source": "/users/gg/mqCBA/definition/provision.xml",
  "workflow-definition-file-original-timestamp": "2016-11-18T20:03:47Z",
  "workflow-id": "ProvisionQueueManager",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0.1",
  "workflow-variable-input-file": "definition/workflow_variables.properties",
  "workflow-variable-input-file-original-source":
    "/users/gg/mqCBA/definition/workflow_variables.properties",
  "workflow-variable-input-file-original-timestamp": "2016-11-18T20:00:42Z",
  "prompt-variables": [],
  "public-variables":
    ["CSQ_CHIN_SERVICE_CLASS_NAME", "CSQ_MSTR_SERVICE_CLASS_NAME", "CSQ_TCPIP_PORT_NUMBER",
     "CSQ_AUTO_GEN_CMD_PFX_SSID", "CSQ_CMD_PFX_FOR_AUTO_GEN", "CSQ_CHIN_REPORT_CLASS_NAME",
     "CSQ_MSTR_CLASSIFICATION_RULE_ID", "CSQ_MSTR_REPORT_CLASS_NAME", "CSQ_CMD_PFX", "CSQ_QSGDISP",
     "CSQ_CHIN_CLASSIFICATION_RULE_ID", "CSQ_TCPIP_STATUS_CODE", "CSQ_TARG_LIB_HLQ", "CSQ_SSID",
     "CSQ_TCPIP_PORT_ID", "CSQ_LANG_LETTER", "CSQ_ENVIRONMENT"]
  ],
  "at-create-variables": [],
  "workflow-clean-after-provisioned": true,
  "security-wf-info": null,
  "create-time": "2016-11-18T20:00:43.504Z",
  "created-by-user": "domadmin",
  "last-modified-by-user": "domadmin",
  "last-modified-time": "2016-11-18T20:04:50.913Z",
  "admin-documentation-file-original-source": "/users/gg/mqCBA/documentation/admin-
mqaas_readme.pdf",
  "admin-documentation":
    "/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
admin",
  "admin-documentation-type": "pdf",
  "consumer-documentation-file-original-source":
    "/users/gg/mqCBA/documentation/consumer-workflow_variables.properties",
  "consumer-documentation":
    "/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
consumer",
  "consumer-documentation-type": "text",
  "base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
  "admin-documentation-mime-type": "application/pdf",
  "consumer-documentation-mime-type": "text/plain",
  "SAF-resources": [],
  "runAsUsers": [],
  "runAsUser-audit": true,
  "automatic-security": true,
  "published-timestamp": "2017-04-05T16:16:55.878Z",
  "archived-timestamp": "",
  "provisioning-version": "1400"
}
```

Get a published software service template history

Use this operation to retrieve the history for a published software service template.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/history
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.
The following value is valid: 1.0.

<existing-entry-name>

Identifies the published template for which history is to be retrieved.

Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

domain-name

Optional, string, specifies the domain name.

Description

This operation retrieves the history for a published software service template.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 302](#).

Request content

None.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

See [“Authorization requirements” on page 49](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the published software service template. [Table 221 on page 303](#) lists the fields in the history object.

Table 221. Response from a get request: History object

Field	Type	Valid for Template Type	Description
action-type	String	Standard, Composite	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"> • Create • Add approval • Approve • Archive • Modify • Publish • Refresh • Reject • Remove approval • Run • Test run • Security complete • Update approval
user	String	Standard, Composite	The user who performed the action.
action-time	String	Standard, Composite	The time that the action was taken.
action-details	String	Standard, Composite	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

Example HTTP interaction

In Figure 114 on page 303, a request is submitted to retrieve the history for a published software service template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/template1/history
```

Figure 114. Sample request to retrieve a published software template history

The following figure shows the response body for the sample request in the previous example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:15.791Z",
      "action-details": "Created template"
    },
    {
      "action-type": "Publish",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:24.860Z",
      "action-details": "Published template"
    }
  ]
}
```

```
} ]
```

Get consumer documentation for a published software service template

Use this operation to retrieve the consumer documentation for a published software service template from the catalog.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/documentation/  
consumer
```

In this request:

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<existing-entry-name>

Identifies the software services template to be retrieved.

documentation/consumer

Causes the consumer documentation file to be retrieved.

Query parameters

None.

Description

This operation retrieves the consumer documentation for a published software service template from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the consumer documentation for a software services template being retrieved.

Request content

None.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

None.

Example HTTP interaction

In [Figure 115 on page 305](#), a request is submitted to retrieve consumer documentation for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2/documentation/consumer
```

Figure 115. Sample request to get consumer documentation for a software services template

Get prompt variables for a published software service template

Use this operation to retrieve the name/value pairs for variables that are required to run the software services template and for which a prompt can be used to obtain the value.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/  
prompt-variables
```

In this request:

<existing-entry-name>

Identifies the software services template for which variables are to be retrieved.

<version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

You can specify the following query parameter on this request to limit the software services templates that are returned. To be returned, a software services template must all query parameters.

domain-name

Optional, specifies the domain name.

Description

This operation retrieves the variables for which a prompt can obtain the value.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 306](#)

Request content

None.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the prompt variables. [Table 222 on page 307](#) lists the fields in the JSON object.

Table 222. Response from a *get prompt variables* request

Field	Type	Template Type	Description
prompt-variables	Array of objects	Standard	An array of required prompt variable objects. See Table 223 on page 307 .
composite-prompt-variables	Array of objects	Composite	Array of composite-prompt-variables objects that contains information about the variables that are expected to be prompted for in preparation for running the composite software services template. See Table 224 on page 308 .

Table 223. Response from a *get request: Prompt-Variable Object*

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Table 224. Response from a get request: Composite-Prompt-Variable Object

Field	Type	Description
published-template-name	String	The name of the published template in the composite template that the prompt-variables field is associated with.
prompt-variables	Array of objects	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the published-template-name software services template as part of the composite software services template. See Table 223 on page 307 .

Example HTTP interaction

Figure 116 on page 308 shows a request to retrieve the prompt variables for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/mq/prompt-variables
```

Figure 116. Sample request to get the prompt variables for a published software service template

The following is the response body for the request.

```

{
  "prompt-variables": [
    {
      "name": "CSQ_MQ_SSID",
      "label": "MQ_SSID",
      "description": "The name of the MQ subsystem to be provisioned.",
      "type": "string",
      "value": "ZCT1",
      "required": true,
      "choices": null,
      "regex": "[A-Z0-9]{1,4}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Subsystem identifier",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "CSQ_CMD_PFX",
      "label": "CMD_PFX",
      "description": "The MQ subsystem command prefix.",
      "type": "string",
      "value": "!ZCT1",
      "required": true,
      "choices": null,
      "regex": "[!\\sa-zA-Z0-9.,!?()*+|=|;%_?:$@#&<>]{1,8}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Command prefix",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "CSQ_ENVIRONMENT",
      "label": "ENVIRONMENT",
      "description": "The environment for which the queue manager is to be provisioned/de-provisioned.  
The BSDS, Log and Pageset datasets vary depending on the environment.",
      "type": "string",
      "value": "TEST",
      "required": true,
      "choices": [
        "DEV",
        "TEST",
        "QA",
        "PROD"
      ],
      "regex": null,
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Environment for which the queue manager is to be provisioned/de-provisioned  
(DEV, TEST, QA, PROD)",
      "multi-line": false,
      "must-be-choice": true,
      "error-message": "The value entered is not valid."
    }
  ]
}

```

Figure 117 on page 309 shows a request to retrieve the prompt variables for a published composite template.

```
GET /zosmf/provisioning/rest/1.0/psc/scout/prompt-variables
```

Figure 117. Sample request to retrieve prompt variables, composite template

The following is the response body for the request.

```

{
  "composite-prompt-variables": [
    {
      "prompt-variables": [],
      "published-template-name": "s1"
    },
    {
      "prompt-variables": [
        {
          "name": "CMD",
          "label": "CMD",
          "description": "CMD",
          "type": "string",
          "value": "S BCTEST",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "CMD",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        },
        {
          "name": "WELSHIE",
          "label": "name",
          "description": "This variable contains the name of a welsh springer
spaniel.",
          "type": "string",
          "value": "Scout",
          "required": false,
          "choices": null,
          "regex": ".*",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "Name of a Welsh Springer Spaniel",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": ".*"
        },
        {
          "name": "INS",
          "label": "INS",
          "description": "INS",
          "type": "string",
          "value": "Instructions",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "INS",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        }
      ],
      "published-template-name": "s2"
    }
  ]
}

```

Figure 118. Response body for the GET prompt variables request, composite template

List the published software service templates

Use this operation to list the software services templates in the catalog that are in the published state.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/psc/
```

In this request, the URI path variable *<version>* identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

Query parameters

You can specify the following query parameter on this request to limit the software services templates that are returned. To be returned, a software services template must all query parameters.

name

Optional, regular expression, specifies the external name of the software services template.

owner

Optional, specifies the user ID or group ID that identifies the owner of the software services template.

software-type

Optional, specifies the type of software being provisioned.

domain-name

Optional, specifies the domain name.

template-type

Optional, specifies the type (standard or composite).

Description

This operation retrieves software services templates that are in the published state from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in software services templates being retrieved. A response body is provided, as described in [“Response content” on page 311](#).

Request content

None.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services templates. See [Table 225 on page 312](#) and [Table 226 on page 312](#).

Table 225. Array of objects

Field	Type	Description
pssc-list	Array of objects	Array of software services template objects. The array is filtered based on any query parameters that were provided.

Table 226. Fields for each software services template

Field	Type	Value Returned for Template Type	Description
generated-name	String	Standard, Composite	The generated name for the software services template.
object-id	String	Standard, Composite	The ID that identifies the software services template.
base-object-id	String	Standard	The object ID that is associated with all the versions of the template. software services template.
name	String	Standard, Composite	Descriptive name for the software services template. The name must be unique. It can be up to 100 characters. The name cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
version	String	Standard	Version of the software services template.
owner	String	Standard, Composite	User ID of the software services template owner.
state	String	Standard, Composite	Indicates the current status of the software services template: published The entry is locked and visible in the marketplace.
description	String	Standard, Composite	Description of the software services template.
domain-name	String	Standard, Composite	Name of the domain this template resides in.
action-definition-file	String	Standard	Location of the action definition file.
software-id	String	Standard	A short, arbitrary, value that identifies the software being provisioned.
software-name	String	Standard	Name of the software being provisioned.
software-type	String	Standard	Identifies the type of software being provisioned.
software-version	String	Standard	Version of the software being provisioned.

Table 226. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
workflow-definition-file	String	Standard	Location of the workflow definition file for the software services template. This file is the primary XML file for the workflow definition.
workflow-id	String	Standard	Identifies the workflow.
workflow-vendor	String	Standard	Name of the vendor that provided the workflow definition file.
workflow-version	String	Standard	Version of the workflow definition file.
workflow-variable-input-file	String	Standard	Optional properties file that specifies values for one or more of the variables that are defined in the workflow definition file.
create-time	String	Standard, Composite	The time that this software services template was created, in ISO 8601 format.
create-by-user	String	Standard, Composite	The user that created this software services template.
last-modified-time	String	Standard, Composite	The last time this software services template was updated, in ISO 8601 format.
last-modified-by-user	String	Standard, Composite	The user that last updated this software services template.
template-type	String	Standard, Composite	Identifies the type of template: standard Defines a single software service. composite Consists of multiple published templates that are provisioned together.
composite-definition	Array of objects	Composite	An array of objects that define the composite template. Not valid if template-type is standard. See Table 227 on page 314 .
composite-variable-input-file	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files. The variable names are in the following format: <standard-template-name>.<atcreate-variable> For example: CICS.startup=10

Table 226. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
composite-parents	Array of strings	Standard	An array of strings. Each string is a composite template that includes this standard template. For example: [c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]
composite-cluster	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
provisioning-version	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Standard, Composite	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.

Table 227. Contents of composite-definition object

Field	Type	Required/optional	Description
sequence	integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
missing	boolean	Required	<ul style="list-style-type: none"> • true if no published template is available that is related to the original version used when the template was defined • false if a published template exists that satisfies the published template requirement
description	String	Required	Description of the software services template.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
software-type	String	Required	Type of software that is being provisioned.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 228. Response from a software services template request failure

Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

Example HTTP interaction

Figure 119 on page 315 shows a request to retrieve a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc HTTP/1.1
```

Figure 119. Sample request to list all published software service templates

The following is the response body for the request.

```
{
  "psc-list": [
    {
      "name": "mqCBA",
      "version": "1",
      "owner": "domadmin",
      "state": "published",
      "description": "This workflow provisions an MQ for z/OS Queue Manager",
      "generated-name": "mqCBA.1.default",
      "object-id": "5b0c3367-b856-4727-99ac-f9a79c9abf28",
      "base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
      "domain-name": "default",
      "action-definition-file": "definition/qmgrActions.xml",
      "software-id": "5655-W97",
      "software-name": "IBM MQ for z/OS",
      "software-type": "QMGR",
      "software-version": "V8.0.0",
      "workflow-definition-file": "definition/provision.xml",
      "workflow-id": "ProvisionQueueManager",
      "workflow-vendor": "IBM",
      "workflow-version": "1.0.1",
      "workflow-variable-input-file": "definition/workflow_variables.properties",
      "create-time": "2016-11-18T20:00:43.504Z",
      "created-by-user": "domadmin",
      "last-modified-by-user": "domadmin",
      "last-modified-time": "2016-11-18T20:28:43.951Z",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    }
  ]
}
```

Figure 120. Response body for the GET request

Modify a published software service template

Use this operation to modify fields in a published software services template.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
```

In this request:

<version>

Is the URI path variable `<version>` that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

<name>

Identifies the published software service template to be modified.

Query parameters

None.

Description

This operation modifies fields in a published software service template, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 316.

Note: Any existing instances that are already created from the template are not affected if this service is used to modify the jobs disposition or workflow disposition.

Request content

The request content is expected to contain a JSON object as described in [Table 229](#) on page 316.

Table 229. Request content for the modify software services template request			
Field name	Type	Valid for template type	Description
description	String	Standard, Composite	Description of the software services template (up to 500 characters).
workflows-disposition	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete. The default is archive.
jobs-disposition	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete. The default is keep.
instances-disposition	String	Standard, Composite	Disposition of instances after they deprovision successfully: keep or delete. The default is keep.

Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements”](#) on page 49.

HTTP status codes

For the valid HTTP status codes, see [“HTTP status codes”](#) on page 284.

Response content

On successful completion, HTTP status code 204 (Normal) is returned, indicating that the request resulted in a modified published software service template.

Example HTTP interaction

In [Figure 121 on page 317](#), a request is submitted to modify a published software service template named `bringUpDB2`.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2
{
  "description": "New description text here",
  "jobs-disposition": "delete",
  "workflows-disposition": "archive"
  "instances-disposition": "keep"
}
```

Figure 121. Sample request to run a software services template

Software services instance services

The software services instance services are application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services instances in the software services registry.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 46.

The software services registry contains a list of the software on z/OS that has been registered as being provisioned, typically through the use of software services templates, although provisioning can be done manually. It is maintained in the z/OSMF data repository and has a sysplex-wide scope.

Table 230 on page 319 lists the operations that the software services instance services provide.

Table 230. z/OSMF software services instance services: operations summary	
Operation name	HTTP method and URI path
“Create a software services instance” on page 321	POST /zosmf/provisioning/rest/<version>/scr
“Get the contents of a software services instance” on page 334	GET /zosmf/provisioning/rest/<version>/scr/<object-id>
“Get the variables for a software services instance” on page 354	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/variables
“Get key-value variables for a software services instance” on page 357	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/key-value-variables
“List the software services instances” on page 345	GET /zosmf/provisioning/rest/<version>/scr
“Update a software services instance” on page 359	PUT /zosmf/provisioning/rest/<version>/scr/<object-id>
“Delete a software services instance” on page 367	DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>
“Perform an action against a software services instance” on page 371	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action>
“Resume a provisioning workflow” on page 369	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/resume-workflow
“Resume an action workflow” on page 374	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/resume-workflow

Table 230. z/OSMF software services instance services: operations summary (continued)	
Operation name	HTTP method and URI path
“Retry a provisioning workflow” on page 376	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/retry-workflow
“Retry an action workflow” on page 378	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/retry-workflow
“Update variables in a software services instance” on page 364	PUT /zosmf/provisioning/rest/version/scr/object-id/variables
“Get the response for an action performed against a software services instance” on page 380	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
“List the responses for actions performed against a software services instance” on page 385	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions
“Delete the response for an action performed against a software services instance” on page 388	DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>

Composite software services instances

Composite software services instances include parent and child software services instances. For more information, see [“Composite templates” on page 199](#).

Authorization requirements

Use of the software services instance services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the domain administrator. The specific requirements for each software services instance service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 49](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 231. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.

Table 231. Response from a request failure (continued)

Field	Type	Description
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Error logging

Errors from the software services instance services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request succeeded and resulted in the creation of an object.

HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

HTTP 204 No content

The request succeeded, but no content is available to be returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 404 Not found

The requested resource does not exist.

HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

Create a software services instance

You can use this operation to create a software services instance in the registry.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software services instance service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates a software services instance in the registry, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 322.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new software services instance. The URI path for the software object is provided in the Location response header and a response body is provided, as described in [“Response content”](#) on page 330.

Request content

The request content is expected to contain a JSON object. [Table 232 on page 322](#) lists the fields in the JSON object.

Table 232. Request content for a create software services instance request			
Field name	Type	Required or optional	Description
type	String	Required	Type of the software. Up to 8 characters.
registry-type	String	Required	The type of software registry object: catalog or general. An object with registry-type catalog is created from a software services template in the software services catalog. When the registry type is catalog, a catalog object ID and catalog name are also required.
state	String	Required	The current state of the software: <ul style="list-style-type: none">• being-initialized• being-provisioned• provisioned• provisioning-suspended• being-deprovisioned• deprovisioning-suspended• deprovisioned• provisioning-failed• deprovisioning-failed
catalog-object-id	String	Required when registry-type is catalog	The identifier of the software services template to be used to create the software services instance.
template-owner	String	Optional	The owner of the template that the registry instance was created from. This field is not valid when the value for the registry-type field is general.
template-version	String	Optional	The version of the software services template to be used to create the software services instance.
catalog-object-name	String	Required when registry-type is catalog	The name of the software services template to be used to create the software services instance.

Table 232. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
external-name	String	Optional	The external name to identify the software registry object. If the external name is not provided then it is set from object-name in the response body. Up to 34 characters.
system-nickname	String	Optional	The nickname of the system the service is provisioned on and where corresponding actions will be run.
system	String	Optional	System on which the software is provisioned. Up to 8 characters.
sysplex	String	Optional	Sysplex on which the software is provisioned. Up to 8 characters.
vendor	String	Optional	Vendor of the software. Up to 24 characters.
version	String	Optional	Version of the software. Up to 24 characters.
description	String	Optional	Description for the software. Up to 256 characters.
owner	String	Optional	The user ID that identifies the owner of the software registry object. Up to 8 characters.
provider	String	Optional	The user ID that identifies the provider of the software, . Up to 8 characters. This is the owner of the software catalog object.
quality-attributes	String	Optional	The quality attributes associated with the software. Up to 16 characters.
workflow-key	String	Optional	The workflow key associated with provisioning the software. This field is not valid when the value for registry-type is general.
workflow-clean-after-provisioned	String	Optional	The indication of whether the workflow instance used to provision this instance will be removed after the workflow is completed. Must be archive, true, or false. The default is archive. This field is not valid when the value for registry-type is general.
job-statement	String	Optional	The JOB statement.
jobs-disposition	String	Optional	Indicates the disposition of jobs. keep Keep all completed jobs located on the JES spool from the provisioning workflow and all action workflows. delete Delete all completed jobs located on the JES spool from the provisioning workflow and all action workflows. The default is keep.

Table 232. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
instances-disposition	String	Optional	Indicates the disposition of the software instance after it is deprovisioned. keep Keep the instance after it is deprovisioned. delete Delete the instance after it is deprovisioned. The default is keep.
actions	Action[]	Optional	The actions for the software. See Table 233 on page 326 .
variables	Variable[]	Optional	The variables for the software. See Table 234 on page 327 .
user-data-id	String	Optional	The user data ID.
user-data	String	Optional	The user data.
domain-id	String	See description.	The domain ID. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
tenant-id	String	See description.	The tenant ID. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
domain-name	String	See description.	The name of the domain. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
tenant-name	String	See description.	The name of the tenant. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
ssin	String	Optional	Software service instance name, used in generating names for software services instances. This field is not valid when the value for registry-type is general.

Table 232. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
runAsUser-audit	boolean	Optional	<p>Indicates if auditing is performed on workflows and command actions that are associated with the instance.</p> <p>false z/OSMF performs validation to ensure that the runAsUser user ID is a valid MVS user ID. No further authorization checking is done prior to switching to the runAsUser ID.</p> <p>true Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow might fail.</p> <p>The default is false.</p> <p>This field is valid only when the registry-type is catalog.</p>
composite-data	Array of objects	Required if the instance is a parent of a composite	<p>If set, indicates that this instance is the parent of a composite. Specifies an array of composite data objects. Each object represents information about an existing catalog type registry instance (registry-type is catalog) that is a child member of this composite software services instance.</p> <p>See Table 236 on page 328.</p>
composite-cluster	String	Optional	<p>Indicates if the instance is either a composite cluster parent or a member, as follows:</p> <p>true The instance is either a composite cluster parent or a member.</p> <p>false The instance is neither a composite cluster parent nor a member.</p>
composite-parent-template-name	String	Optional	Name of the template for the composite parent.
composite-parent-template-id	String	Optional	ID of the template for the composite parent.
expiration-period	String	Optional	Number of days the software services instance is kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0".

Table 233. Action structure for a "create a software services instance" request

Field	Type	Description
name	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software. You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.
type	String	The type of the action. The value must be one of the following: <ul style="list-style-type: none"> • command • workflow • instructions
is-deprovision	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> • If true, the action deprovisions the software. • If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
command	String	For command type actions, the command.
command-run-as-user	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
command-sol-key	String	For command type actions, if provided, the key to search for in the solicited messages command response.
command-unsol-key	String	For command type actions, if provided, the key to search for in the unsolicited messages.
command-detect-time	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages. Also, the minimum time before a command response is checked for after the command is submitted for execution. If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
workflow-definition-file	String	For workflow type actions, the workflow definition file.
workflow-variable-input-file	String	For workflow type actions, if provided, the workflow variable input file.
variables	Variable[]	For workflow type actions, if provided, the workflow variables.

Table 233. Action structure for a "create a software services instance" request (continued)

Field	Type	Description
workflow-clean-after-complete	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
instructions	String	For instruction type actions, the instructions.
prompt-variables	PromptVariable[]	For workflow type actions, if provided, the prompt variables, which are the variables that will have their values prompted for at create time. See Table 235 on page 327 .
at-create-variables	String[]	For workflow type actions, if provided, the names of the at create variables, which are the only variables allowed on input-variables for the do action operation.
command-run-as-user-dynamic	boolean	Indicates if the command-run-as-user ID value can change: true The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content. false The command-run-as-user ID is final and cannot change during the processing of the command.

Table 234. Variable structure

Field	Type	Description
name	String	Name of the variable.
description	String	Description of the variable.
value	String	Value of the variable.
visibility	String. The value must be public or private.	Visibility of the variable.
update-registry	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 235. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.

Table 235. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Table 236. Composite-data structure

Field	Type	Required/ optional	Description
object-id	String	Required	Object ID that is associated with the existing child instance.
sequence	Integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of objects	Optional	<p>An array of connector object.</p> <p>The connector variables specified here take precedence over the variables field in this object and any variables that are specified in the workflow variable input file that is associated with the published template.</p> <p>See Table 237 on page 329.</p>

Table 236. Composite-data structure (continued)

Field	Type	Required/ optional	Description
variables	Array of objects	Optional	<p>A list of one or more variables for the provisioning workflow that is associated with this published template. The variables specified here take precedence over the variables that are specified in the workflow variable input file.</p> <p>Specify this property as an array of name-value objects, for example:</p> <pre>"variables": [{"name": "user_name", "value": "IBMUSER"}, {"name": "file_name", "value": "textfile.txt"}]</pre>
workflow-definition-file	String	Required	The absolute path for the provisioning workflow definition file.
variable-input-file	String	Optional	The absolute path for the variable input file that is associated with the provisioning workflow definition file.

Table 237. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.
not-valid	boolean	Required	<p>Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is:</p> <ul style="list-style-type: none"> • false, if all of the information is accurate • true, if one or more of the values are incorrect.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 330](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 238. HTTP error response codes for a create software services instance request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the service returns the following:

- URI path of the created software services instance in the Location response header.
- Response body, which contains a JSON object with details about the software services instance. [Table 239 on page 330](#) lists the fields in the JSON object.

Table 239. Response from a create software services instance request		
Field	Type	Description
object-id	String	The object ID of the newly created object. The object ID is to be used on further requests to the object.
object-uri	String	The object URI of the newly created object.
object-name	String	The object name of the newly created object.
external-name	String	The external name of the newly created object.
system-nickname	String	Nickname of the system that the service is provisioned on.
cluster-instance-name	String	The name of the cluster instance that this entry belongs to.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 240. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.

<i>Table 240. Response from a request failure (continued)</i>		
Field	Type	Description
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interactions

In [Figure 122 on page 332](#), a request is submitted to create a software services instance on the system SY1.

```

{
  "type": "DB2",
  "external-name": "DB2B",
  "vendor": "IBM",
  "version": "V5R10",
  "description": "DB2 for test1",
  "registry-type": "catalog",
  "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
  "catalog-object-name": "DB2template1",
  "template-owner": "ZOSMFAD",
  "workflow-key": "02e1ec78-e0db-482b-8013-3d435b52e2e3",
  "workflow-clean-after-provision": "true",
  "system-nickname": "SYSTEM1",
  "system": "SY1",
  "sysplex": "PLEX1",
  "state": "being-provisioned",
  "owner": "ZOSMFAD",
  "provider": "ZOSMFAD",
  "quality-attributes": "123456789ABCDEF0",
  "user-data": "my data",
  "user-data-id": "udid1",
  "domain-id": "izu$0",
  "tenant-id": "izu$002",
  "domain-name": "default",
  "tenant-name": "default",
  "ssin": "SSIN1",
  "composite-cluster": "true",
  "composite-parent-template-id": "2ed65dd8-7c4a-4029-8e37-576714df38ee",
  "composite-parent-template-name": "c5",
  "variables": [
    {
      "name": "IACTION_NAME",
      "value": "Instructions1",
      "visibility": "public"
    },
    {
      "name": "COMMAND1",
      "value": "d a,l",
      "visibility": "public"
    },
    {
      "name": "C_DETECT_TIME",
      "value": "25",
      "visibility": "public"
    },
    {
      "name": "C_SOL_K",
      "value": "VLF",
      "visibility": "public"
    },
    {
      "name": "C_UNSQL_K",
      "value": "CSV",
      "visibility": "public"
    }
  ],
  "actions": [
    {
      "name": "Instructions1",
      "type": "instructions",
      "is-deprovision": "false",
      "instructions": "These are the instructions for the ${IACTION_NAME} action."
    },
    {
      "name": "command1",
      "type": "command",
      "is-deprovision": "false",
      "command": "${COMMAND1}",
      "command-detect-time": "${C_DETECT_TIME}",
      "command-run-as-user": "IBMUSER",
      "command-sol-key": "${C_SOL_K}",
      "command-unsol-key": "${C_UNSQL_K}"
    },
    {
      "name": "deprovision",
      "type": "instructions",
      "is-deprovision": "true",
      "instructions": "Do the deprovision manually."
    }
  ]
}

```

Figure 122. Sample request to create a software services instance

The response is shown below.

```

{
  "object-name": "DB2_1",
  "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
  "object-uri": "/zosmf/provisioning/rest/1.0/scr/c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
  "external-name": "DB2_DY01",
  "cluster-instance-name": "Y0",

```

```
"system-nickname": "SY1"
}
```

In Figure 123 on page 333, a request is submitted to create a composite software services instance.

```
{
  "type": "forCics",
  "description": "composite for Cics",
  "registry-type": "catalog",
  "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
  "catalog-object-name": "cics_comp",
  "state": "being-provisioned",
  "domain-id": "izu0",
  "tenant-id": "izu002",
  "domain-name": "default",
  "tenant-name": "default",
  "composite-data": [
    {
      "sequence": 1,
      "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
      "published-template-name": "mq",
      "connectors": [],
      "variables": [
        { "name": "user_name", "value": "IBMUUSER" },
        { "name": "defect_status", "value": "approved" }
      ],
      "deprovisioning-action": "deprovision_2",
      "workflow-definition-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/mq.1.default/definition/p.xml"
    },
    {
      "sequence": 2,
      "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94fA",
      "published-template-name": "cics",
      "connectors": [
        {
          "variable-name": "cics_mq_ssn",
          "source-template": "mq",
          "source-variable-name": "mq_ssn"
        }
      ],
      "variables": [
        { "name": "defect_status", "value": "approved" }
      ],
      "deprovisioning-action": "deprovision_2",
      "workflow-definition-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/cics.1.default/definition/p.xml",
      "variable-input-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/cics.1.default/definition/var.properties"
    }
  ]
}
```

Figure 123. Sample request for a composite

Get the contents of a software services instance

You can use this operation to retrieve the contents of a software services instance.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.

Query parameters

None.

Description

This operation retrieves the properties of a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 242 on page 335](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 242 on page 335](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 241. HTTP error response codes for a get software services instance contents request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found; the software services instance does not exist.

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 242](#) on page 335 lists the fields in the JSON object.

Table 242. JSON object that is returned for a get software services instance property request

Field	Type	Description
object-id	String	The object-id for the software services instance
object-name	String	The object-name for the software services instance
type	String	Type of the software. The value is null for composite parent registry instances.
registry-type	String	Type of registry object: catalog or general
external-name	String	External name of the software services instance
system-nickname	String	The nickname of the system that the software is provisioned on.
system	String	System that the software is provisioned on.
sysplex	String	Sysplex that the software is provisioned on.
last-known-system	String	The name of the system on which the software for the instance was last known to be running.
last-known-system-nickname	String	The nickname of the system on which the software for the instance was last known to be running.
vendor	String	Vendor of the software. Null for a composite instance.
version	String	Version of the software. Null for a composite instance.
description	String	Description for the software
owner	String	The user ID that identifies the owner of the software
provider	String	The user ID that identifies the provider of the software
template-owner	String	The owner of the template that the registry instance was created from.
template-version	String	The version of the software services template used when provisioning the software represented by this instance. The template-version field can be null or empty if the instance was created before support for the template-version field was added.
catalog-object-id	String	The identifier of the template that is used when provisioning the software represented by this instance. Only valid when registry-type is catalog.
catalog-object-name	String	The name of the template that was used when provisioning the software represented by this instance.

Table 242. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
workflow-key	String	The workflow key that is associated with provisioning the software.
workflow-clean-after-provisioned	String	The indication of whether the workflow instance used to provision this instance is removed after the workflow is completed. Must be archive, true, or false. The default is archive. This field is not valid when the value for registry-type is general.
job-statement	String	The JOB statement.
jobs-disposition	String	<p>Indicates the disposition of jobs.</p> <p>keep Keep all completed jobs located on the JES spool from the provisioning workflow and all action workflows.</p> <p>delete Delete all completed jobs located on the JES spool from the provisioning workflow and all action workflows.</p> <p>The default is keep.</p>
instances-disposition	String	<p>Indicates the disposition of the software instance after it is deprovisioned.</p> <p>keep Keep the instance after it is deprovisioned.</p> <p>delete Delete the instance after it is deprovisioned.</p> <p>The default is keep.</p>
state	String	<p>The current state of the software:</p> <ul style="list-style-type: none"> • being-initialized • being-provisioned • provisioned • provisioning-suspended • being-deprovisioned • deprovisioning-suspended • deprovisioned • provisioning-failed • deprovisioning-failed
quality-attributes	String	The quality attributes of the software
actions	Action[]	The actions for the software. Table 243 on page 339
variables	Variable[]	The variables for the software. See Table 244 on page 341 .

Table 242. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
workflow-start-time	String	The time that workflow processing started, in ISO8601 format. The value is null if the workflow was not started.
workflow-stop-time	String	The time that workflow automation last stopped, in ISO8601 format. The value is null if the workflow automation has not stopped.
workflow-current-step-name	String	The name of the step that is being processed automatically in the provisioning workflow or action workflow. This field is set if workflow automation is in progress or stopped.
workflow-current-step-number	String	The number of the step that is being processed automatically in the provisioning workflow or action workflow. This field is set if workflow automation is in progress or stopped.
workflow-total-steps	String	The total number of steps in the workflow.
created-time	String	The time the object was created. The time is in the ISO8601 format.
last-modified-time	String	The time the object was updated. The time is in the ISO8601 format. This field might be updated during creation, and the value might be later than the value for the created-time field.
expiration-period	String	Number of days the software services instance will be kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0". This field is optional.
has-expired	String	Indicates whether the provisioned instance is expired ("true" or "false").
expiration-date	String	The date and time the instance expires. The date is in the ISO8601 format. By default, this value is null. This field is optional.
expiration-upcoming-notified	String	The time the consumer was notified of the upcoming instance expiration. The time is in the ISO8601 format.
expiration-process-notified	String	The time the consumer was notified that the instance expired. The time is in ISO8601 format.
expiration-processed-reminder-notified	String	The time the consumer and domain administrators were last sent a reminder that the instance is expired. The time is in ISO8601 format.
created-by-user	String	The user ID that created the object

Table 242. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
last-modified-by-user	String	The user ID that last updated the object
last-action-name	String	The name of the last action that was performed.
last-action-object-id	String	The action ID of the last action that was performed.
last-action-state	String	The state of the last action that was performed.
user-data-id	String	The user data ID.
user-data	String	The user data.
domain-id	String	The domain ID.
tenant-id	String	The tenant ID.
domain-name	String	The name of the domain.
tenant-name	String	The name of the tenant.
ssin	String	Software service instance name, used in generating names for software services instances.
runAsUser-audit	boolean	<p>Indicates if auditing is performed on workflows and command actions that are associated with the instance.</p> <p>false z/OSMF performs validation to ensure that the runAsUser user ID is a valid MVS user ID. No further authorization checking is done prior to switching to the runAsUser ID.</p> <p>true Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow might fail.</p> <p>The default is false.</p> <p>This field is valid only when the registry-type is catalog.</p>
composite-instance-data	Array of objects	<p>Array of composite instance data objects. Each object represents information about an existing catalog type registry instance that is a member of this registry instance. This includes various properties of that instance.</p> <p>See Table 246 on page 342.</p>

Table 242. JSON object that is returned for a get software services instance property request (continued)		
Field	Type	Description
composite-parent-object-id	String	If set, indicates that this instance is a child member of a composite. Specifies the object ID of the registry instance that is the parent of the composite.
composite-children	Array of objects	If set, indicates that this instance is the parent of a composite. Specifies an array of composite child objects. Each object contains information about an existing catalog type registry instance that is a child member of this composite software services instance. See Table 247 on page 342 .
object-uri	String	The object URI of the instance.
workflow-message-text	String	If set, the message text that is associated with the provisioning workflow.
account-information	String	The account information.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
composite-cluster	String	Indicates whether the instance is either a composite cluster parent or a member, as follows: true The instance is either a composite cluster parent or a member. false The instance is not a composite cluster parent or a member.
cluster-instance-name	String	Name of the cluster instance.
composite-parent-template-name	String	Name of the template for the composite parent.
composite-parent-template-id	String	ID of the template for the composite parent.

Table 243. Action structure for a "get the contents of a software services instance" request		
Field	Type	Description
name	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software. You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.

Table 243. Action structure for a "get the contents of a software services instance" request (continued)

Field	Type	Description
type	String	The type of the action. The value must be one of the following: <ul style="list-style-type: none"> • command • workflow • instructions
is-deprovision	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> • If true, the action deprovisions the software. • If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
command	String	For command type actions, the command.
command-run-as-user	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
command-sol-key	String	For command type actions, if provided, the key to search for in the solicited messages command response.
command-unsol-key	String	For command type actions, if provided, the key to search for in the unsolicited messages.
command-detect-time	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages. Also, the minimum time before a command response is checked for after the command is submitted for execution. If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
workflow-definition-file	String	For workflow type actions, the workflow definition file.
workflow-variable-input-file	String	For workflow type actions, if provided, the workflow variable input file.
variables	Variable[]	For workflow type actions, if provided, the workflow variables.
workflow-clean-after-complete	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
instructions	String	For instruction type actions, the instructions.

Table 243. Action structure for a "get the contents of a software services instance" request (continued)

Field	Type	Description
prompt-variables	PromptVariable[]	For workflow type actions, if provided, the prompt variables, which are the variables that are expected to be prompted for in preparation for running the software services template. See Table 245 on page 341.
at-create-variables	String[]	For workflow type actions, if provided, these are the names of the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both. These are the only variables allowed on input-variables for the do action operation.
command-run-as-user-dynamic	boolean	Indicates if the command-run-as-user ID value can change: true The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content. false The command-run-as-user ID is final and cannot change during the processing of the command.
email	String	Email.

Table 244. Variable structure

Field	Type	Description
name	String	Name of the variable.
description	String	Description of the variable.
value	String	Value of the variable.
visibility	String. The value must be public or private.	Visibility of the variable.
update-registry	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 245. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.

Table 245. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Table 246. Composite-instance data

Field	Type	Required/ optional	Description
sequence	Integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
object-id	String	Required	Object ID that is associated with the existing child instance.
catalog-object-name	String	Optional	The published template name that is associated with the child software service instance.
state	String		The current state of the child software service instance.
variables	Variable[]		The variables for the child software service instance. See Table 244 on page 341 .
actions	Action[]		The actions for the child software service instance. See Table 243 on page 339 .
workflow-key	String		The workflow key that is associated with provisioning the child software service instance.

Table 247. Composite child object

Field	Type	Description
sequence	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
object-id	String	Object ID that is associated with the existing child instance.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 248. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the GET method is used to retrieve information about a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: dbc0dfe1-15f4-44aa-ac72-c138ac19155a.

```
GET /zosmf/provisioning/rest/ 1.0/scr/dbc0dfe1-15f4-44aa-ac72-c138ac19155a
```

Figure 124. Sample request to get software services instance properties

The following is an example of the response.

```
{
  "object-id": "dbc0dfe1-15f4-44aa-ac72-c138ac19155a",
  "object-name": "TEST_2",
  "object-uri": "/zosmf/provisioning/rest/1.0/scr/dbc0dfe1-15f4-44aa-ac72-c138ac19155a",
  "type": "TEST",
  "registry-type": "catalog",
  "external-name": "TEST_PLEX1_GP02",
  "system-nickname": "SY1",
  "last-known-system-nickname": "SY1",
  "last-known-system": "SY1",
  "system": "SY1",
  "sysplex": "PLEX1",
  "vendor": "IBM",
  "version": "Version 1",
  "description": null,
  "owner": "ibmuser",
  "provider": "ibmuser",
  "catalog-object-id": "590364bd-1281-4459-95d5-1fa9d20e2f66",
  "catalog-object-name": "template2",
  "state": "provisioned",
  "quality-attributes": null,
  "workflow-key": "8d0b63ce-961c-4099-9b95-30bbdce35d00",
  "workflow-clean-after-provisioned": "archive",
  "jobs-disposition": "keep",
  "instances-disposition": "keep",
  "variables": [
    {
      "name": "VAR1",
      "value": "Value",
      "visibility": "private",
      "update-registry": null
    }
  ],
  "actions": [
    {
      "name": "deprovision",
      "description": "",
      "type": "workflow",
      "is-deprovision": "true",
```

```

        "command": null,
        "command-run-as-user": null,
        "command-run-as-user-dynamic": false,
        "command-sol-key": null,
        "command-unsol-key": null,
        "command-detect-time": null,
        "workflow-definition-file": ".IzuScc-template2.2.domain1/definition/d.xml",
        "workflow-variable-input-file": null,
        "variables": null,
        "workflow-clean-after-complete": "inherit",
        "instructions": null,
        "prompt-variables": [],
        "at-create-variables": null
    }
},
"created-time": "2022-04-08T12:52:46.617Z",
"last-modified-time": "2022-04-08T12:52:55.148Z",
"created-by-user": "ibmuser",
"last-modified-by-user": "ibmuser",
"last-action-name": null,
"last-action-object-id": null,
"last-action-state": null,
"user-data": null,
"user-data-id": null,
"tenant-id": "IYU100",
"tenant-name": "tenant1",
"domain-id": "IYU1",
"domain-name": "domain1",
"ssin": "GP02",
"email": null,
"job-statement": "",
"account-info": null,
"runAsUser-audit": "true",
"workflow-start-time": "2022-04-08T12:52:52.387Z",
"workflow-stop-time": "2022-04-08T12:52:52.424Z",
"composite-children": null,
"composite-instance-data": null,
"composite-parent-object-id": "25ac1cdd-5c8d-400c-906e-614e3d9cfabe",
"provisioning-version": "1800",
"workflow-message-text": "IZUWF0162I: Automation processing for workflow
\\TEST_PLEX1_GP02provision1649422372117\\ is complete.",
"template-owner": "ibmuser",
"workflow-current-step-name": "",
"workflow-current-step-number": null,
"workflow-total-steps": "1",
"composite-cluster": "false",
"composite-parent-template-id": "a5056015-72ec-410a-968f-a4a952f1dd64",
"composite-parent-template-name": "composite1",
"cluster-instance-name": null,
"has-expired": "false",
"expiration-date": null,
"expiration-period": "0",
"expiration-upcoming-notified": "",
"expiration-processed-notified": "",
"expiration-processed-reminder-notified": "",
"template-version": "2"
}

```

List the software services instances

You can use this operation to list the software services instances in the software services registry.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr
```

In this request, the URI path variable is as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.

Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

type

Optional, specifies the type of the software.

registry-type

Optional, specifies the type of the registry object: Catalog or general.

object-name

Optional, regular expression, specifies the name for the software services instance.

external-name

Optional, regular expression, specifies the external name of the software.

system

Optional, specifies the system on which the software is provisioned.

sysplex

Optional, specifies the sysplex on which the software is provisioned.

vendor

Optional, specifies the vendor of the software.

owner

Optional, specifies the user ID that identifies the owner of the software.

provider

Optional, specifies the user ID that identifies the provider of the software.

state

Optional, specifies the current state of the software:

- being-initialized
- being-provisioned
- provisioned
- provisioning-suspended
- being-deprovisioned
- deprovisioning-suspended
- deprovisioned
- provisioning-failed
- deprovisioning-failed

catalog-object-id

Optional, specifies the catalog object ID associated with the creation of this software services instance.

user-data-id

Optional, specifies the ID for the user data.

domain-id

Optional, specifies the ID of the domain.

tenant-id

Optional, specifies the ID of the tenant.

tenant-name

Optional, regular expression that specifies the name of the tenant.

domain-name

Optional, regular expression that specifies the name of the domain.

If you specify no query parameters, then all objects are returned.

Description

This operation returns the object ID and a subset of fields for software services instances.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 346](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For catalog registry type objects, the user that issues the request must be at least one of the following for a software services instance to be returned in the list:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 346](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 249. HTTP error response codes for a list software services instances request</i>	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 401 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the response body contains a JSON object, named `scr-list`, of registry objects that consist of a subset of the fields for all software services instances matching the query. [Table 250 on page 347](#) lists the fields in the JSON object.

Table 250. JSON object that is returned for a list software services instances request

Field	Type	Description
object-id	String	The object ID for the software services instance.
object-name	String	The object name for the software services instance.
type	String	The type of the software, for example, the subsystem name, such as Db2 or IBM MQ. Not returned for composite parent registry instances.
registry-type	String	Type of registry object: Catalog or general.
external-name	String	External name of the software services instance.
template-owner	String	The owner of the template that the registry instance was created from.
template-version	String	The version of the software services template used when provisioning the software represented by this instance. The template-version field can be null or empty if the instance was created before support for the template-version field was added.
catalog-object-id	String	The identifier of the software services template used when provisioning the software represented by this software services instance. Valid only when the value for registry-type is catalog.
catalog-object-name	String	The name of the template that was used when provisioning the software represented by this software services instance.
system	String	Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member..
sysplex	String	Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility..
system-nickname	String	The nickname of the system, indicating the system and sysplex on which the instance was provisioned.
last-known-system	String	The name of the system on which the software for the instance was last known to be running.
last-known-system-nickname	String	The nickname of the system on which the software for the instance was last known to be running.
vendor	String	Vendor of the software.
version	String	Version of the software.
description	String	Description for the software.
owner	String	The user ID that identifies the owner of the software.
provider	String	The user ID that identifies the provider of the software.

Table 250. JSON object that is returned for a list software services instances request (continued)

Field	Type	Description
state	String	The current state of the software: <ul style="list-style-type: none"> • being-initialized • being-provisioned • provisioned • provisioning-suspended • being-deprovisioned • deprovisioning-suspended • deprovisioned • provisioning-failed • deprovisioning-failed
created-time	String	The time the object was created. The time is in the ISO8601 format.
last-modified-time	String	The time the object was updated. The time is in the ISO8601 format.
expiration-period	String	Number of days the software services instance will be kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0". This field is optional.
has-expired	String	Indicates whether the provisioned instance is expired ("true" or "false").
expiration-date	String	The date and time the instance expires. The date is in the ISO8601 format. By default, this value is null. This field is optional.
expiration-upcoming-notified	String	The time the consumer was notified of the upcoming instance expiration. The time is in the ISO8601 format.
expiration-process-notified	String	The time the consumer was notified that the instance expired. The time is in ISO8601 format.
expiration-processed-reminder-notified	String	The time the consumer and domain administrators were last sent a reminder that the instance is expired. The time is in ISO8601 format.
created-by-user	String	The user ID that created the object.
last-modified-by-user	String	The user ID that last updated the object.
last-action-name	String	The name of the last action that was performed.
last-action-object-id	String	The action ID of the last action that was performed.
last-action-state	String	The state of the last action that was performed.
user-data-id	String	The user data ID.
tenant-id	String	The tenant ID.
domain-id	String	The domain ID.
tenant-name	String	The name of the tenant.
domain-name	String	The name of the domain.

Table 250. JSON object that is returned for a list software services instances request (continued)		
Field	Type	Description
composite-children	Array of objects	If set, indicates that this instance is the parent of a composite. Specifies an array of composite child objects. Each object contains information about an existing catalog type registry instance that is a child member of this composite software services instance. See Table 251 on page 349 .
composite-parent-object-id	String	If set, indicates that this instance is a child member of a composite. Specifies the object ID of the registry instance that is the parent of the composite.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> • true if the operations are allowed • false if the operations are not allowed.
composite-cluster	String	Indicates whether the instance is either a composite cluster parent or a member, as follows: <p>true The instance is either a composite cluster parent or a member.</p> <p>false The instance is not a composite cluster parent or a member.</p>
cluster-instance-name	String	Name of the cluster instance.

Table 251. Composite child object		
Field	Type	Description
sequence	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
object-id	String	Object ID that is associated with the existing child instance.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 252. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.

Table 252. Response from a request failure (continued)

Field	Type	Description
debug	String	Debug information about for the error.

Example HTTP interaction

In Figure 125 on page 350, the GET method is used to list the software services instances.

```
GET /zosmf/provisioning/rest/ 1.0/scr
```

Figure 125. Sample request to list software services instances

“Sample response from a list software services instances request” on page 350 shows the response.

Sample response from a list software services instances request

```
{
  "scr-list": [
    {
      "object-id": "a877c442-0a33-43e9-9899-7e1a91ff6752",
      "object-name": "TEST_1",
      "external-name": "TEST_PLEX1_GP01",
      "system": "SY1",
      "sysplex": "PLEX1",
      "type": "TEST",
      "registry-type": "catalog",
      "catalog-object-id": "17322e4b-f3b1-403d-b53b-df5e773575e5",
      "catalog-object-name": "template1",
      "vendor": "IBM",
      "version": "Version 1",
      "owner": "ibmuser",
      "provider": "ibmuser",
      "description": null,
      "state": "provisioned",
      "created-time": "2022-04-08T12:52:46.544Z",
      "last-modified-time": "2022-04-08T12:52:50.069Z",
      "created-by-user": "ibmuser",
      "last-modified-by-user": "ibmuser",
      "last-action-name": null,
      "last-action-object-id": null,
      "last-action-state": null,
      "user-data-id": null,
      "tenant-id": "IYU100",
      "tenant-name": "tenant1",
      "domain-id": "IYU1",
      "domain-name": "domain1",
      "composite-children": null,
      "composite-parent-object-id": "25ac1cdd-5c8d-400c-906e-614e3d9cfabe",
      "last-known-system-nickname": "SY1",
      "system-nickname": "SY1",
      "last-known-system": "SY1",
      "provisioning-version": "1800",
      "provisioning-version-supported": true,
      "template-owner": "ibmuser",
      "composite-cluster": "false",
      "cluster-instance-name": null,
      "instance-managed-by": null,
      "has-expired": "false",
      "expiration-date": null,
      "expiration-period": "0",
      "expiration-upcoming-notified": "",
      "expiration-processed-notified": "",
      "expiration-processed-reminder-notified": "",
      "template-version": "1"
    },
    {
      "object-id": "dbc0dfe1-15f4-44aa-ac72-c138ac19155a",
      "object-name": "TEST_2",
      "external-name": "TEST_PLEX1_GP02",
      "system": "SY1",

```

```

"sysplex": "PLEX1",
"type": "TEST",
"registry-type": "catalog",
"catalog-object-id": "590364bd-1281-4459-95d5-1fa9d20e2f66",
"catalog-object-name": "template2",
"vendor": "IBM",
"version": "Version 1",
"owner": "ibmuser",
"provider": "ibmuser",
"description": null,
"state": "provisioned",
"created-time": "2022-04-08T12:52:46.617Z",
"last-modified-time": "2022-04-08T12:52:55.148Z",
"created-by-user": "ibmuser",
"last-modified-by-user": "ibmuser",
"last-action-name": null,
"last-action-object-id": null,
"last-action-state": null,
"user-data-id": null,
"tenant-id": "IYU100",
"tenant-name": "tenant1",
"domain-id": "IYU1",
"domain-name": "domain1",
"composite-children": null,
"composite-parent-object-id": "25ac1cdd-5c8d-400c-906e-614e3d9cfabe",
"last-known-system-nickname": "SY1",
"system-nickname": "SY1",
"last-known-system": "SY1",
"provisioning-version": "1800",
"provisioning-version-supported": true,
"template-owner": "ibmuser",
"composite-cluster": "false",
"cluster-instance-name": null,
"instance-managed-by": null,
"has-expired": "false",
"expiration-date": null,
"expiration-period": "0",
"expiration-upcoming-notified": "",
"expiration-processed-notified": "",
"expiration-processed-reminder-notified": "",
"template-version": "2"
},
{
  "object-id": "25ac1cdd-5c8d-400c-906e-614e3d9cfabe",
  "object-name": "GP_1",
  "external-name": "GP_PLEX1_GP00",
  "system": "SY1",
  "sysplex": "PLEX1",
  "type": null,
  "registry-type": "catalog",
  "catalog-object-id": "a5056015-72ec-410a-968f-a4a952f1dd64",
  "catalog-object-name": "composite1",
  "vendor": null,
  "version": null,
  "owner": "ibmuser",
  "provider": "ibmuser",
  "description": null,
  "state": "provisioned",
  "created-time": "2022-04-08T12:52:46.417Z",
  "last-modified-time": "2022-04-08T12:52:57.406Z",
  "created-by-user": "ibmuser",
  "last-modified-by-user": "ibmuser",
  "last-action-name": null,
  "last-action-object-id": null,
  "last-action-state": null,
  "user-data-id": null,
  "tenant-id": "IYU100",
  "tenant-name": "tenant1",
  "domain-id": "IYU1",
  "domain-name": "domain1",
  "composite-children": [
    {
      "sequence": 1,
      "object-id": "a877c442-0a33-43e9-9899-7e1a91ff6752"
    },
    {
      "sequence": 2,
      "object-id": "dbc0dfe1-15f4-44aa-ac72-c138ac19155a"
    }
  ],
  "composite-parent-object-id": null,
  "last-known-system-nickname": "SY1",

```

```

    "system-nickname": "SY1",
    "last-known-system": "SY1",
    "provisioning-version": "1800",
    "provisioning-version-supported": true,
    "template-owner": "ibmuser",
    "composite-cluster": "false",
    "cluster-instance-name": null,
    "instance-managed-by": null,
    "has-expired": "false",
    "expiration-date": null,
    "expiration-period": "0",
    "expiration-upcoming-notified": "",
    "expiration-processed-notified": "",
    "expiration-processed-reminder-notified": "",
    "template-version": "1"
  }
]
}

```


Get the variables for a software services instance

You can use this operation to retrieve the variables for a software services instance.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/variables
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.

Query parameters

You can specify the following query parameter on this request:

name

Use this optional parameter to specify variable names. You can use regular expressions.

visibility

Use this optional parameter to specify the visibility of the variables.

Description

This operation retrieves the variables for a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 355](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, , if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 355](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 253. HTTP error response codes for a get software services instance variables request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.

Table 253. HTTP error response codes for a get software services instance variables request (continued)	
HTTP error status code	Description
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Response content

On successful completion, the response body contains a JSON object consisting of an array of the variable names and values for the software services instance, described in the tables that follow.

Table 254. Get variables request: Format of the variables object		
Field name	Type	Description
variables	Array of variables	Variables for the software services instance. Refer to Table 255 on page 355

Table 255. Variable structure		
Field	Type	Description
name	String	Name of the variable.
description	String	Description of the variable.
value	String	Value of the variable.
visibility	String. The value must be public or private.	Visibility of the variable.
update-registry	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 256. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the GET method is used to retrieve variables for a software object. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
GET /zosmf/provisioning/rest/1.0/76963ea5-81a4-42d6-99d6-f3e19747cf61/variables
```

Figure 126. Sample request to get software services instance variables

An example of the response is shown in the figures that follow.

```
{
  "variables": [
    {
      "name": "INS",
      "description": "This is some text that describes the variable.",
      "value": "Instructions",
      "visibility": "public",
      "update-registry": "false"
    }
  ]
}
```

Figure 127. Sample response from a get software services instance variables request

Get key-value variables for a software services instance

You can use this operation to retrieve the variables for a software services instance in key-value format.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/key-value-variables
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance to be retrieved.

Query parameters

You can specify the following query parameter on this request:

name

Use this optional parameter to specify variable names. You can use regular expression.

Description

This operation retrieves the variables for a z/OSMF software services instance in key-value format.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 358](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: **<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES**.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (normal) is returned and the response body is provided, as described in [“Response content” on page 358](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 257. HTTP error response codes for a get software services instance key-value variables request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance instance was not found; the software services instance does not exist.

Response content

On successful completion, the response body contains a JSON object that consists of an array of the variable names and values for the software services instance, described in the tables that follow.

Table 258. Get key-value variables request: Format of the variables object		
Field name	Type	Description
variables	List of variable names and values in key-value pair format. Example: <pre>{ "var1": "val1", "var2": "val2" }</pre>	Variables for the software services instance. Only variables with public visibility are returned.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 259. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the GET method is used to retrieve key-value variables for a software object. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
GET /zosmf/provisioning/rest/1.0/76963ea5-81a4-42d6-99d6-f3e19747cf61/key-value-variables
```

Figure 128. Sample request to get software services instance variables in key-value format

The following is an example of the response.

```
{  
  "variables": {  
    { "var1": "val1",  
      "var2": "val2" }  
  }  
}
```

Figure 129. Sample response from a get key-value variables request

Update a software services instance

You can use this operation to update fields in a software services instance.

HTTP method and URI path

```
PUT /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be updated.

Query parameters

None.

Description

This operation updates a software services instance.

Request content

The request content is expected to contain a JSON object containing the fields to be updated. [Table 260](#) on page 359 lists the fields that are valid.

Table 260. Request content for an update software services instance request			
Field name	Type	Required or optional	Description
state	String	Optional	The current state of the software: <ul style="list-style-type: none">• being-initialized• being-provisioned• provisioned• provisioning-suspended• being-deprovisioned• deprovisioning-suspended• deprovisioned• provisioning-failed• deprovisioning-failed
external-name	String	Optional	The external name to identify the software registry object. Up to 34 characters.
system	String	Optional	System on which the software is provisioned, up to eight characters. Cannot be updated if the registry type is catalog for the software services instance.

Table 260. Request content for an update software services instance request (continued)

Field name	Type	Required or optional	Description
sysplex	String	Optional	Sysplex on which the software is provisioned, up to eight characters Cannot be updated if the registry type is catalog for the software services instance.
vendor	String	Optional	Vendor of the software, up to 24 characters Cannot be updated if the registry type is catalog for the software services instance.
version	String	Optional	Version of the software, up to 24 characters Cannot be updated if the registry type is catalog for the software services instance.
description	String	Optional	Description for the software, up to 256 characters
owner	String	Optional	The user ID that identifies the owner of the software registry object, up to eight characters Cannot be updated if the registry type is catalog for the software services instance.
provider	String	Optional	The user ID that identifies the provider of the software, up to eight characters. This is the owner of the software catalog object. Cannot be updated if the registry type is catalog for the software services instance.
quality-attributes	String	Optional	The quality attributes associated with the software, up to 16 characters Cannot be updated if the registry type is catalog for the software services instance.
workflow-key	String	Optional	The workflow key associated with provisioning the software. This field is not valid when the value for registry-type is general.
workflow-clean-after-provisioned	String	Optional	The indication of whether the workflow instance used to provision this instance will be removed after the workflow is completed. Must be true or false. This field is not valid when the value for registry-type is general.
actions	Action[]	Optional	The actions for the software. Cannot be updated if the registry type is catalog for the software services instance. See Table 261 on page 361 .

Table 260. Request content for an update software services instance request (continued)

Field name	Type	Required or optional	Description
variables	Variable[]	Optional	The variables for the software. Refer to Table 262 on page 362 . Cannot be updated if the registry type is catalog for the software services instance.
user-data-id	String	Optional	The user data ID.
user-data	String	Optional	The user data.
ssin	String	Optional	Software service instance name, used in generating names for software services instances. This field is not valid when the value for registry-type is general.

Table 261. Action structure for an "update software services instance" request

Field	Type	Description
name	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software. You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.
type	String Must be one of the following values: <ul style="list-style-type: none"> • command • workflow • instructions 	The type of the action.
is-deprovision	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> • If true, the action deprovisions the software. • If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
command	String	For command type actions, the command.
command-run-as-user	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
command-sol-key	String	For command type actions, if provided, the key to search for in the solicited messages command response.
command-unsol-key	String	For command type actions, if provided, the key to search for in the unsolicited messages.

Table 261. Action structure for an "update software services instance" request (continued)

Field	Type	Description
command-detect-time	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages. Also, the minimum time before a command response is checked for after the command is submitted for execution. If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
workflow-definition-file	String	For workflow type actions, the workflow definition file.
workflow-variable-input-file	String	For workflow type actions, if provided, the workflow variable input file.
variables	Variable[]	For workflow type actions, if provided, the workflow variables. See Table 262 on page 362.
instructions	String	For instruction type actions, the instructions.
workflow-clean-after-complete	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
prompt-variables	PromptVariable[]	Prompt variables, for workflow type actions, if any are provided. At create time, there are prompts for the values. See Table 263 on page 362.
at-create-variables	String	Names of the at create variables, for workflow type actions, if any are provided. These are the only variables that are allowed on input variables for the do action operation.

Table 262. Variable structure

Field	Type	Description
name	String	Name of the variable.
description	String	Description of the variable.
value	String	Value of the variable.
visibility	String. The value must be public or private.	Visibility of the variable.
update-registry	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 263. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.

Table 263. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

The user issuing the request must be one of the following:

- Owner of the software services instance
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the provisioning administrator.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 Normal is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 264. HTTP error response codes for a update software services instance request

HTTP error status code	Description
HTTP 400 Bad request	Request contained incorrect parameters.
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance instance was not found; the software services instance does not exist.
HTTP 409	The field cannot be updated for the registry type.

Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 265. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the PUT method is used to update a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
PUT /zosmf/provisioning/rest/ 1.0/scr/ 76963ea5-81a4-42d6-99d6-f3e19747cf61
```

```
{
  "state": "provisioned"
}
```

Figure 130. Sample request to update a software services instance property

Update variables in a software services instance

You can use this operation to update variables in a software services instance.

HTTP method and URI path

```
PUT /zosmf/provisioning/rest/version/scr/object-id/variables
```

In this request, the URI path variables are described, as follows:

- `<version>` identifies the version of the provisioning service. The following value is valid: 1.0.
- `<object-id>` identifies the software services instance for which variables are to be updated.

Query parameters

None.

Description

This operation updates variables in a software services instance. If it already exists, the value and visibility are updated based on the values in the variable structure.

Request content

The request body contents must be a JSON object containing a variables field with the variables to be updated in the object. See [Table 266 on page 365](#).

Table 266. Request content for the update software services instance variables request		
Field name	Type	Description
variables	Variable[]	The variables for the software, with the structure that is described in Table 267 on page 365 . The name field identifies the variable. If a variable in the variables array does not already exist in the software object it is added. If it does already exist the name and visibility are updated based on the values in the variable structure.

Table 267. Variable structure		
Field	Type	Description
name	String	Name of the variable.
description	String	Description of the variable.
value	String	Value of the variable.
visibility	String. The value must be public or private.	Visibility of the variable.
update-registry	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

The user issuing the request must be one of the following:

- Owner of the software services instance
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the provisioning administrator.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 Normal is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 268. HTTP error response codes for a update software services instance request	
HTTP error status code	Description
HTTP 400 Bad request	Request contained incorrect parameters.
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance instance was not found.

Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 269. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the PUT method is used to update variables for a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
PUT /zosmf/provisioning/rest/1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61/variables
```

Variable structure:

```
{
  "variables": [
    { "name": "var1", "value": "val1", "visibility": "public" },
    { "name": "var2", "value": "val2", "visibility": "public" }
  ]
}
```

Figure 131. Sample request to update variables for a software services instance

Delete a software services instance

The delete operation removes a software services instance from the software services registry.

HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be deleted.

Query parameters

None.

Description

This operation removes a z/OSMF software services instance. The state of a software services instance must be one of the following:

- deprovisioned
- deprovisioning-failed
- provisioning-failed.

Deleting a composite software services instance deletes all of the associated child instances.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

The user issuing the request must be the owner of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 *Normal* is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

<i>Table 270. HTTP error response codes for a delete software services instance request.</i>	
deletion of a parent software service instance will delete all the corresponding children instances.	
deletion of a parent software service instance will delete all the corresponding children instances.	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.
HTTP 409 Request conflict	The software services instance could not be removed because its state was not either deprovisioned, deprovisioning-failed, or provisioning-failed.

Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 271. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the DELETE method is used to delete a software services instance. The software services instance is uniquely identified by a key, which is the string value 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
DELETE /zosmf/provisioning/rest/ 1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61
```

Figure 132. Sample request to delete a software services instance

Resume a provisioning workflow

You can use this operation to resume a provisioning workflow that is suspended.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/resume-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.

Query parameters

None.

Description

This operation resumes a provisioning workflow that is suspended.

On successful completion, HTTP status code 204 (Normal) is returned.

Request content

None.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

The user issuing the request must be the owner of the software object, or a domain administrator of the software object.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 272. HTTP error response codes for a resume provisioning workflow request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.
HTTP 409 Conflict	A conflict exists.

Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 273. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the POST method is used to resume a provisioning workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/resume-workflow
```

Figure 133. Sample request to resume a provisioning workflow for a software services instance

The response is 204. There is no response body.

Perform an action against a software services instance

You can use this operation to perform an action against a software services instance.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance.
- **<action>** identifies the action to be performed.

Query parameters

None.

Description

This operation performs an action against a software services instance.

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 372](#).

Note: You cannot deprovision the child instances of a composite instance (that is, an instance created from a composite template). Instead, deprovision against the composite instance.

Request content

The request content is expected to contain a JSON object. See [Table 274 on page 371](#) for the fields.

Table 274. Request content for the perform action software services instance request			
Field name	Type	Required or optional	Description
input-variables	input Variable[]	Optional	The input variables to be used by workflow-type actions. See Table 275 on page 371 .
target-system-nickname	String	Optional	The system nickname indicating the system in the sysplex to run the action on.

Table 275. Input variable structure	
Field	Type
name	String
value	String

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

The user issuing the request must be one of the following:

- Owner of the software services instance

- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the provisioning administrator.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 372](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

<i>Table 276. HTTP error response codes for a do action request</i>	
HTTP error status code	Description
HTTP 400 Error	Bad request.
HTTP 403 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Response content

On successful completion, the response body contains a JSON object consisting of the response from the action.

<i>Table 277. Response body for the do action request</i>		
Field name	Type	Description
action-id	String	The ID of the action object that was created by running the action. The action ID is used on further requests to the action object.
action-uri	String	The URI of the new action object.

If a failure occurs, the response body contains a JSON object with a description of the error.

<i>Table 278. Response from a request failure</i>		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the POST method is used to perform an action for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/start
```

Figure 134. Sample request to perform an action against a software services instance variables

An example of the response is shown in the figures that follow.

```
{
  "action-id": "65963ea5-81a4-42d6-99d6-f3e19748cf61",
  "action-uri":
    "/zosmf/provisioning/rest/1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61/actions/65963ea5-81a4-42d6-99d6-f3e19748cf61"
}
```

Figure 135. Sample response from a get software services instance variables request

Resume an action workflow

You can use this operation to resume an action workflow that is suspended.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/resume-workflow
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance.
- **<action-id>** identifies the action to be resumed.

Query parameters

None.

Description

This operation resumes an action workflow that is suspended.

On successful completion, HTTP status code 204 (Normal) is returned.

Request content

None.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: **<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES**.

The user issuing the request must be one of the following:

- Owner of the software object
- Domain administrator of the software object
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 279. HTTP error response codes for a resume action workflow request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Table 279. HTTP error response codes for a resume action workflow request (continued)

HTTP error status code	Description
HTTP 409 Conflict	A conflict exists.

Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 280. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the POST method is used to resume an action workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9/
resume-workflow
```

Figure 136. Sample request to resume an action workflow for a software services instance

The response is 204. There is no response body.

Retry a provisioning workflow

You can use this operation to restart a failed provisioning workflow.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/retry-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.

Query parameters

None.

Description

This operation restarts a provisioning workflow that failed. The workflow instance is restarted at the workflow step that failed.

On successful completion, HTTP status code 204 (Normal) is returned.

Request content

None.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

The user issuing the request must be the owner of the software object, or a domain administrator of the software object.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 281. HTTP error response codes for a retry provisioning workflow request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.
HTTP 409 Conflict	A conflict exists.

Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 282. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the POST method is used to retry a provisioning workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/retry-workflow
```

Figure 137. Sample request to retry a provisioning workflow for a software services instance

The response is 204. There is no response body.

Retry an action workflow

You can use this operation to restart a failed action workflow.

HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/retry-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.
- *<action-id>* identifies the action to be retried.

Query parameters

None.

Description

This operation restarts an action workflow that failed. The workflow instance is restarted at the workflow step that failed.

On successful completion, HTTP status code 204 (Normal) is returned.

Request content

None.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

The user issuing the request must be one of the following:

- Owner of the software object
- Domain administrator of the software object
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 283. HTTP error response codes for a retry action workflow request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Table 283. HTTP error response codes for a retry action workflow request (continued)

HTTP error status code	Description
HTTP 409 Conflict	A conflict exists.

Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 284. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the POST method is used to retry an action workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9/
retry-workflow
```

Figure 138. Sample request to retry an action workflow for a software services instance

The response is 204. There is no response body.

Get the response for an action performed against a software services instance

You can use this operation to retrieve information about the response for an action that was performed against a software services instance.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.
- *<action-id>* identifies the actions object to be retrieved.

Query parameters

None.

Description

This operation retrieves an action object that describes the response for an action that was performed against a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 286 on page 381](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant of the software services instance, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 286 on page 381](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 285. HTTP error response codes for a get software services instance contents request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Table 285. HTTP error response codes for a get software services instance contents request (continued)	
HTTP error status code	Description
HTTP 404 Not found	The specified software services instance was not found; the software services instance does not exist.

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 286](#) on [page 381](#) lists the fields in the JSON object.

Table 286. JSON object that is returned for a get actions request		
Field	Type	Description
action-id	String	The action ID for the action object.
name	String	The name for the action.
type	String	Type of the action.
is-deprovision	String must be: <ul style="list-style-type: none"> • true • false 	If true, the action is a deprovision action. Otherwise, the action is not a deprovision action.
state	String must be: <ul style="list-style-type: none"> • in-progress • submitted • suspended • responded • warning • complete • failed 	The current state of the action. The values submitted, responded, and warning are valid only for command type actions. The suspended value is valid only for workflow type actions. For the command action state of warning, see the command-response, command-sol-key-hit, and command-detection-status fields. Either no response was received, the command-sol-key-hit is false, or the command-detection-status is expired.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
ran-at-time	String	The time the do action operation was done to create the action, in ISO8601 format
ran-by-user	String	The user ID that ran the do action operation that created the action
instructions	String	The instructions associated with the action, or null if no instructions are associated
command	String	The command associated with the action, or null if no command is associated
command-response	String	The solicited messages response from the command.
command-sol-key-hit	String: null, true, or false	If the command-sol-key was specified, indicates whether the command-sol-key was found in the solicited message response. If the command-sol-key-hit is false then the action state is set to warning.

Table 286. JSON object that is returned for a get actions request (continued)

command-detection-message	String	If the command-unsol-key was specified and it was found in the unsolicited messages from the command, the message containing the command-unsol-key.
command-detection-status	String: null, waiting, expired, or detected	If the command-unsol-key was specified, this is the status of whether the command-unsol-key was found in the unsolicited messages. If the command-detection-status is expired then the action state is set to warning.
workflow-key	String	The workflow key of the workflow associated with the action, or null if no workflow is associated
workflow-current-step-name	String	The current workflow step name of the workflow associated with the action, or null
workflow-message-id	String	The workflow message ID for the workflow associated with the action, or null
workflow-message-text	String	The workflow message text for the workflow associated with the action, or null
workflow-name	String	The workflow name for the workflow associated with the action, or null
workflow-status-name	String	The workflow status name for the workflow associated with the action, or null.
workflow-start-time	String	The time that workflow processing started, in ISO8601 format. The value is null if the workflow was not started.
workflow-stop-time	String	The time that workflow automation last stopped, in ISO8601 format. The value is null if the workflow automation has not stopped.
system	String	System that the software is provisioned on.
sysplex	String	Sysplex that the software is provisioned on.
system-nickname	String	The nickname of the system that the software is provisioned on.
composite-parent-action-id	String	The action ID for the composite parent's action that is associated with this action.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 287. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.

Table 287. Response from a request failure (continued)

Field	Type	Description
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the GET method is used to retrieve the response for an action that was performed for a software services instance.

```
GET /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions/764190f2-350b-4d08-b036-f3bb0a861068
```

Figure 139. Sample request to get software services instance actions

The following is an example of the response.

```
{
  "name": "deprovision",
  "state": "complete",
  "type": "workflow",
  "command": null,
  "instructions": null,
  "action-id": "764190f2-350b-4d08-b036-f3bb0a861068",
  "ran-at-time": "2017-04-17T15:20:07.480Z",
  "ran-by-user": "IBMUSER",
  "is-deprovision": "true",
  "workflow-current-step-name": "",
  "workflow-key": "100f4645-94df-472b-81f2-f8268e2e73e3",
  "workflow-message-id": "IZUWF0162I",
  "workflow-message-text": "IZUWF0162I: Automation processing for workflow \"MIX_DB2000deprovision1492442407521\" is complete.",
  "workflow-name": "MIX_DB2000deprovision1492442407521",
  "workflow-status-name": "complete",
  "command-response": null,
  "command-sol-key-hit": null,
  "command-detection-message": null,
  "command-detection-status": null,
  "workflow-start-time": "2017-04-17T15:20:07.702Z",
  "workflow-stop-time": "2017-04-17T15:20:07.726Z",
  "system": "SYS1",
  "sysplex": "PLEX1",
  "system-nickname": "SY1"
}
```

Figure 140. Sample response for performed actions

List the responses for actions performed against a software services instance

You can use this operation to list the responses for actions that were performed against a software services instance.

HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance for which actions are to be retrieved.

Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

type

Optional, specifies the type of the software.

name

Optional, regular expression, specifies the name of the action object.

state

Optional, specifies the current state of the action:

- in-progress
- submitted
- suspended
- responded
- warning
- complete
- failed.

If you specify no query parameters, all actions are returned.

Description

This operation lists the action objects for actions that were performed against a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 290 on page 386](#).

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES*.

For catalog registry-type objects, the user issuing the request must be at least one of the following:

- The owner of the software object
- A member of the tenant that the software object is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances

- A domain administrator of the software object.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 290 on page 386](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 288. HTTP error response codes for a "get software services instance contents" request</i>	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found; the software services instance does not exist.

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. See [Table 289 on page 386](#) and [Table 290 on page 386](#) lists the fields in the JSON object.

<i>Table 289. JSON object that is returned for a list actions request</i>		
Field	Type	Description
scr-list-actions	Array of objects	Array of action objects.

<i>Table 290. Action object for a list actions request</i>		
Field	Type	Description
action-id	String	The action ID for the action object.
name	String	The name for the action.
type	String	Type of the action.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
state	String. Must be one of the following: <ul style="list-style-type: none"> • in-progress • submitted • suspended • responded • warning • complete • failed. 	The current state of the action. The values submitted, responded, and warning are valid only for command type actions. The suspended value is valid only for workflow type actions.
ran-at-time	String	The time the do action operation was done to create the action, in ISO8601 format
ran-by-user	String	The user ID that ran the do action operation that created the action

Table 290. Action object for a list actions request (continued)

Field	Type	Description
composite-parent-action-id	String	The action ID for the composite parent's action that is associated with this action.
workflow-key	String	Workflow key of the workflow that is associated with the action, or null if no workflow is associated.
workflow-name	String	Workflow name of the workflow that is associated with the action, or null if no workflow is associated.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 291. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the GET method is used to retrieve the list of responses for actions that were performed against a software services instance.

```
GET /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions
```

Figure 141. Sample request to list performed actions

The following is an example of the response.

```
{
  "scr-list-actions":
  [
    {
      "name": "Instructions1",
      "state": "complete",
      "type": "instructions",
      "action-id": "f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9",
      "ran-at-time": "2015-10-26T18:29:20.949Z",
      "ran-by-user": "ZOSMFAD"
    },
    {
      "name": "deprovision",
      "state": "complete",
      "type": "workflow",
      "action-id": "ae3ec9cc-9be3-42b4-98f5-aa64934e31a3",
      "ran-at-time": "2015-10-27T14:34:27.186Z",
      "ran-by-user": "ZOSMFAD"
    }
  ]
}
```

Figure 142. Sample response from a list actions request

Delete the response for an action performed against a software services instance

The delete operation removes the response for an action that was performed against a software services instance.

HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance for which an action response is to be deleted.
- **<action-id>** identifies the action for which the response is to be deleted.

Query parameters

None.

Description

This operation removes the response for an action that was performed against a software services instance.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: **<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES**.

The user issuing the request must be one of the following:

- Owner of the software services instance
- For catalog registry type objects, one of the following: a domain administrator of the software services instance or a member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- For general registry type objects, the provisioning administrator.

For more information, see [“Authorization requirements” on page 320](#).

HTTP status codes

On successful completion, HTTP status code 204 **Normal** is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 292. HTTP error response codes for a delete action response request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 293. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In the following example, the DELETE method is used to delete the response for an action that was performed against a software services instance.

```
DELETE /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9
```

Figure 143. Sample request to delete a response for a performed action

Software service instance name services

The software service instance name (SSIN) services are application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software service instance names.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 46.

Table 294 on page 390 lists the operations that the SSIN services provide.

Table 294. SSIN services: operations summary	
Operation name	HTTP method and URI path
“Create software service instance names” on page 391	POST /zosmf/resource-mgmt/rest/<version>/ssin
“List the software service instance names” on page 394	GET /zosmf/resource-mgmt/rest/<version>/ssin
“Create a variable name” on page 396	POST /zosmf/resource-mgmt/rest/<version>/ssin/variable-name
“Create unique variable names” on page 398	POST /zosmf/resource-mgmt/rest/<version>/unique-variable-names

Authorization requirements

Use of the SSIN services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

The user’s z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT.

Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 295. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Error logging

Errors from the software services instance services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 Normal

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request succeeded and resulted in the creation of an object.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

Create software service instance names

You can use this operation to create software service instance names (SSINs).

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/ssin
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates SSINs. It uses the name-prefix in the resource definition profile as a basis for creating the names. An initial SSIN is created when a software instance is provisioned. A maximum of 8 generated SSINs may exist in the resource definition profile that the software instance is using. Allocation of SSINs for a provisioned software instance are released when the software instance is deprovisioned. The name-prefix in the resource definition profile must end with the special wildcard character, `*`.

For the properties that you can specify in the request body, a JSON object, see [“Request content” on page 391](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a SSINs.

Request content

The request content is expected to contain a JSON object. [Table 296 on page 392](#) lists the fields in the JSON object.

Table 296. Request content for the create SSIN request			
Field name	Type	Required or optional	Description
template-id	String	Required	The ID of the template.
domain-id	String	Required	The ID of the domain.
tenant-id	String	Required	The ID of the tenant.
registry-id	String	Required	The ID of the software instance registry.
quantity	String	Required	The number of names to be generated. The value must be 1-7.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES.

For more information, see [“Authorization requirements” on page 390](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 392](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 297. HTTP error response codes for a create SSIN request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the service returns a JSON object named ssin-list consisting of the names that were created. See [Table 298 on page 392](#).

Table 298. Response from a create SSIN request		
Field	Type	Description
ssin-list	Array	Software service instance names. See Table 299 on page 392 .

Table 299. Fields in the ssin-list array		
Field	Type	Description
ssin	String	Software service instance name.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 300. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.

Table 300. Response from a request failure (continued)

Field	Type	Description
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In Figure 144 on page 393, a request is submitted to create 2 SSINs.

```
POST /zosmf/resource-mgmt/rest/1.0/ssin
{
  "domain-id": "izu$0",
  "registry-id": "046c3cb2-7ef2-40a0-8b10-a34d8a23e5fc",
  "template-id": "9eb7df8a-284c-4550-a92e-8150bc6fe68f",
  "tenant-id": "izu$000",
  "quantity": "2"
}
```

Figure 144. Sample request to create SSINs

The response body is as follows.

```
{
  "ssin-list": [
    {
      "ssin": "INAME101"
    },
    {
      "ssin": "INAME201"
    }
  ]
}
```

List the software service instance names

You can use this operation to list the software service instance names (SSINs).

HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/ssin
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

name

Name of the object for which SSINs should be obtained.

registry-id

Identifier of the registry for which SSINs should be obtained.

If you specify no query parameters, then all SSINs are returned.

Description

The list operation returns software service instance names based on the input query.

On successful completion, HTTP status code 200 (Normal) is returned, along with a response body.

Request content

None.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT`.

See [“Authorization requirements” on page 390](#).

HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 395](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 301. HTTP error response codes for a list SSIN request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the service returns a JSON object named `ssin-list` consisting of the names that were created. See [Table 302 on page 395](#).

Table 302. Response from a list SSINs request

Field	Type	Description
ssin-list	Array	Software service instance names. See Table 303 on page 395 .

Table 303. Fields in the ssin-list array

Field	Type	Description
ssin	String	Software service instance name.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none">• true if the operations are allowed• false if the operations are not allowed.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 304. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

In [Figure 145 on page 395](#), a request is submitted to list the SSINs for `name=INAME.*`.

```
GET /zosmf/resource-mgmt/rest/1.0/ssin?name=INAME.*
```

Figure 145. Sample request to list SSINs

The response body is as follows.

```
{
  "ssin-list": [
    {
      "ssin": "INAME101",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    },
    {
      "ssin": "INAME201",
```

```

    "provisioning-version": "1400",
    "provisioning-version-supported": true
  }
]
}

```

Create a variable name

You can use this operation to create a variable name.

HTTP method and URI path

Create software service instance names

```
POST /zosmf/resource-mgmt/rest/<version>/ssin/variable-name
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates a variable name based on the input variable prefix and the last 2 digits from the SSIN for the input registry ID. For the properties that you can specify, see [“Request content” on page 396](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new variable name. A response body is provided, as described in [“Response content” on page 397](#).

Request content

The request content contains a JSON object. [Table 305 on page 396](#) lists the fields in the JSON object.

Table 305. Request content for the create variable name request			
Field name	Type	Required or optional	Description
variable-prefix	String	Required	The prefix to use to create the variable name.
registry-id	String	Required	The ID of the software instance registry entry.

Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 390](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 397](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 306. HTTP error response codes for a create variable request

HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the variable name. [Table 307 on page 397](#) lists the fields in the JSON object.

Table 307. Response from a create variable name request

Field	Type	Description
name	String	Variable name.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 308. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

Figure 146 on page 397 shows a request to create a variable name.

```
POST /zosmf/resource-mgmt/rest/1.0/ssin/variable-name
{
  "variable-prefix": "VAR",
  "registry-id": "3196202f-9a6c-4fdf-8dcd-e307e3ce2d5b"
}
```

Figure 146. Sample request to create a variable name

The response body is as follows.

```
{
  "name": "VAR00"
}
```

Create unique variable names

You can use this operation to create multiple unique variable names.

HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/unique-variable-names
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

Query parameters

None.

Description

This operation creates up to 50 unique variable names. For the properties that you can specify, see [“Request content” on page 398](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of unique variable names. A response body is provided, as described in [“Response content” on page 399](#).

Request content

The request content contains a JSON object. [Table 309 on page 398](#) lists the fields in the JSON object.

Table 309. Request content for the create unique variable names request			
Field name	Type	Required or optional	Description
prefix	String	Optional	The prefix to be used when creating the variable names. If a prefix is not specified on the request, one is supplied by the service.
quantity	String	Required	The number of names to be generated, 1-50.

Authorization requirements

None.

For more information, see [“Authorization requirements” on page 390](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 399](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 310. HTTP error response codes for a create variable request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.

Table 310. HTTP error response codes for a create variable request (continued)

HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the variable names. [Table 311 on page 399](#) lists the fields in the JSON object.

Table 311. Response from a create unique variable names request

Field	Type	Description
name-list	String Array	The variable names that were created.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 312. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

Example HTTP interaction

[Figure 147 on page 399](#) shows a request to create 5 unique variable names.

```
POST /zosmf/resource-mgmt/rest/1.0/unique-variable-name
{
  "prefix": "VAR",
  "quantity": "5"
}
```

Figure 147. Sample request to create a variable name

The response body is as follows.

```
{
  "name-list": [
    "VAR1462458322735",
    "VAR1462458322737",
    "VAR1462458322739",
    "VAR1462458322741",
    "VAR1462458322743"
  ]
}
```

Data persistence services

The data persistence services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with user-specific data and global application data, as described in this topic.

Table 313 on page 400 lists the operations that the data persistence services provide.

Table 313. Operations provided through the data persistence services	
Operation	HTTP method and URI path
“Persist user or application data” on page 401	PUT /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue> PUT /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
“Retrieve persisted user or application data” on page 403	GET /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue> GET /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
“Delete persisted user or application data” on page 406	DELETE /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue> DELETE /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>

Required authorizations

The user must be logged in to z/OSMF, and must have READ access to the SAF profile that was registered for the plug-in and task making the request.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all data persistence services as input and output for the requested operations. The attributes provided in the JSON object depend on the requested operation.

```
{
  "value": "data-value",
  "version": "structure-version",
  "messages": "z/OSMF-messages",
  "update": true|false
}
```

where:

data-value

The value that will be added, updated, retrieved, or removed by the data persistence services. Any data type is supported including JSON objects, JSON arrays, and scalars. The value is required.

structure-version

Version of the data persistence services and the JSON object structure used for this request. The version sequence starts at 1.0.0, and is incremented only if the services or the JSON structure changes. The version the client supports is required as input to the request. The data persistence services is backward compatible for $n-2$ versions, and accepts requests for each version it supports. If the version specified by the client is not supported or if no version is specified, the service returns an error message.

z/OSMF-messages

z/OSMF messages received during the request. The *messages* attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

update

An optional input attribute, which indicates that the service is updating or replacing an existing JSON object. If you set the value to *true*, the service updates the key-value pairs you specified for the *value* attribute and preserves any other data persisted in the JSON object. You can set this attribute to *true* only when the data type is a JSON object or JSON array. If you omit this attribute or set it to *false*, the service deletes the existing JSON object and creates a new JSON object that contains only the key-value pairs you specified for the *value* attribute.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400® Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the data persistence services.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the data persistence services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Persist user or application data

You can use this operation to persist data to be used by a specific user or application.

HTTP method and URI path

```
PUT /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>
PUT /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>
?saif=<saifparamValue>
```

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will persist the data only for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will persist the data globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.

- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to persist data for the *history* attribute shown in the sample JSON object in [Figure 148 on page 402](#), specify the following resource path: /SETTINGS/history/.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS0803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

Figure 148. Sample JSON structure for persisted data

Query parameters

saf-parameter

The SAF resource that is defined in the properties file when loading the external plug-in.

Standard headers

Use the following standard HTTP headers with this request:

```
Accept: application/json
Content-Type: application/json
```

Custom headers

None.

Request content

Your request must include a JSON object that contains the value to be persisted and the version. For more details, see [“Content type used for HTTP request and response data” on page 400](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 400](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the current data after being modified. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

To persist data that satisfies the following criteria, submit the request depicted in [Figure 149 on page 403](#):

- The data is for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data is being persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the persistence data uses the structure provided in [Figure 148 on page 402](#).
- The data to be persisted is updating the *rsize* attribute.

```
PUT /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS/history/ HTTP/1.1
Host: zosmf1.yourco.com
Accept: application/json
Content-Type: application/json

{
  "version" : "1.0.0",
  "value" : {"history":{"rsize":["40000"]}},
  "update" : true
}
```

Figure 149. Sample request to persist user-specific data

A sample response is shown in [Figure 150 on page 403](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Content-Type: application/json

{
  "version" : "1.0.0",
  "value" : {
    "authorization":{"auth":true},
    "history":{"ugrp":["ZOSMFGRP"],"acct":["OMVS0803"],"rsize":["40000"],"proc":["CEANNKJ"]},
    "trace":{"init":false,"task":false}
  }
}
```

Figure 150. Sample response from a request to persist user-specific data

Retrieve persisted user or application data

You can use this operation to retrieve data that is persisted for a specific user or application.

HTTP method and URI path

```
GET /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
GET /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
```

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will retrieve the data that has been persisted for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will retrieve the data that has been persisted globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.
- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the persisted data. The persisted data is stored in a JSON object using a tree structure. To retrieve persisted data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to retrieve the data persisted for the *history* attribute shown in the sample JSON object in [Figure 151](#) on page 404, specify the following resource path: `/SETTINGS/history/`. In which case, the value for the *acct*, *proc*, *rsize*, and *ugrp* attributes will be retrieved. To retrieve the value for only the *rsize* attribute, specify the following resource path: `/SETTINGS/history/rsize/`.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS00803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

Figure 151. Sample JSON structure for persisted data

Query parameters

saf-parameter

The SAF resource that is defined in the properties file when loading the external plug-in.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 400](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the retrieved data. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

To retrieve the persisted data that satisfies the following criteria, submit the request depicted in [Figure 152 on page 405](#):

- The data was persisted for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data was persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the data uses the structure provided in [Figure 151 on page 404](#).
- The data persisted for the *SETTINGS* attribute is to be retrieved.

```
GET /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS?saf=ZOSMF.IBMMYAPP.MYTASK HTTP/1.1 HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 152. Sample request to retrieve persisted data

A sample response is shown in [Figure 153 on page 405](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2014 05:39:28 +0000GMT
Connection: close

{
  "value": {
    "authorization": { "auth": true },
    "history": { "ugrp": [ "ZOSMFGRP" ], "acct": [ "OMVS0803" ], "rsize": [ "50000" ], "proc": [ "CEANNKJ" ] },
    "trace": { "init": false, "task": false }
  },
  "version": "1.0.0"
}
```

Figure 153. Sample response from a request to retrieve persisted data

Delete persisted user or application data

You can use this operation to remove data that is persisted for a specific user or application.

HTTP method and URI path

```
DELETE /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
DELETE /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
```

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will delete data that has been persisted for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will delete data that has been persisted globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.
- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the data to be deleted. The persisted data is stored in a JSON object using a tree structure. To delete persisted data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to delete the data persisted for the *history* attribute shown in the sample JSON object in [Figure 154](#) on page 406, specify the following resource path: `/SETTINGS/history/`. In which case, the value for the *acct*, *proc*, *rsize*, and *ugrp* attributes will be deleted. To delete the value for only the *rsize* attribute, specify the following resource path: `/SETTINGS/history/rsize/`.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS0803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

Figure 154. Sample JSON structure for persisted data

Query parameters

saf-parameter

The SAF resource that is defined in the properties file when loading the external plug-in.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 400](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes the updated JSON object. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

To delete the persisted data that satisfies the following criteria, submit the request depicted in [Figure 155 on page 407](#):

- The data was persisted for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data was persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the data uses the structure provided in [Figure 154 on page 406](#).
- The data persisted for the *history* attribute is to be deleted.

```
DELETE /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS/history?  
saf=ZOSMF.IBMMYAPP.MYTASK  
HTTP/1.1 HTTP/1.1  
Host: zosmf1.yourco.com
```

Figure 155. Sample request to delete persisted data

A sample response is shown in [Figure 156 on page 408](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{
  "version" : "1.0.0",
  "value":{
    "authorization":{"auth":true},
    "history":null,
    "trace":{"init":false,"task":false}
  }
}
```

Figure 156. Sample response from a request to delete persisted data

Multisystem routing services

To communicate with and transfer data between systems within your enterprise, z/OSMF uses z/OSMF-to-z/OSMF communication. That is, a z/OSMF instance communicates with other z/OSMF instances to collect information from or about the systems in your enterprise. To enable this capability, each system in your enterprise must be accessible by a z/OSMF instance. Typically, this requires deploying one z/OSMF instance in each monoplex or sysplex in your enterprise.

Although your enterprise can have multiple active z/OSMF instances, it is recommended that you make one instance the primary. The *primary z/OSMF instance* is the instance that:

- Is the base for configuring the other z/OSMF instances in your enterprise.
- Generates the Lightweight Third Party Authentication (LTPA) key that is used for single sign-on (if single sign-on is enabled).
- Is used to perform all z/OS system management tasks in your enterprise, which ensures that the data for each z/OSMF task is centrally managed.
- Acts as the client for all hypertext transfer protocol (HTTP) requests and drives the transfer of files between z/OSMF instances.

You can select any z/OSMF instance that is at least z/OSMF V2R1 with APAR PI32148 to be the primary instance. The remaining z/OSMF instances are referred to as *secondary z/OSMF instances*.

Example

For example, suppose your installation is configured similar to the installation depicted in [Figure 157 on page 409](#). The installation contains three sysplexes with a total of nine running systems. A z/OSMF instance is active in each sysplex, and your web browser is connected to the z/OSMF instance that is running on System 3 in Sysplex A. Thus, this z/OSMF instance is the primary instance and the z/OSMF instance running on System 6 in Sysplex B and System 9 in Sysplex C are the secondary instances.

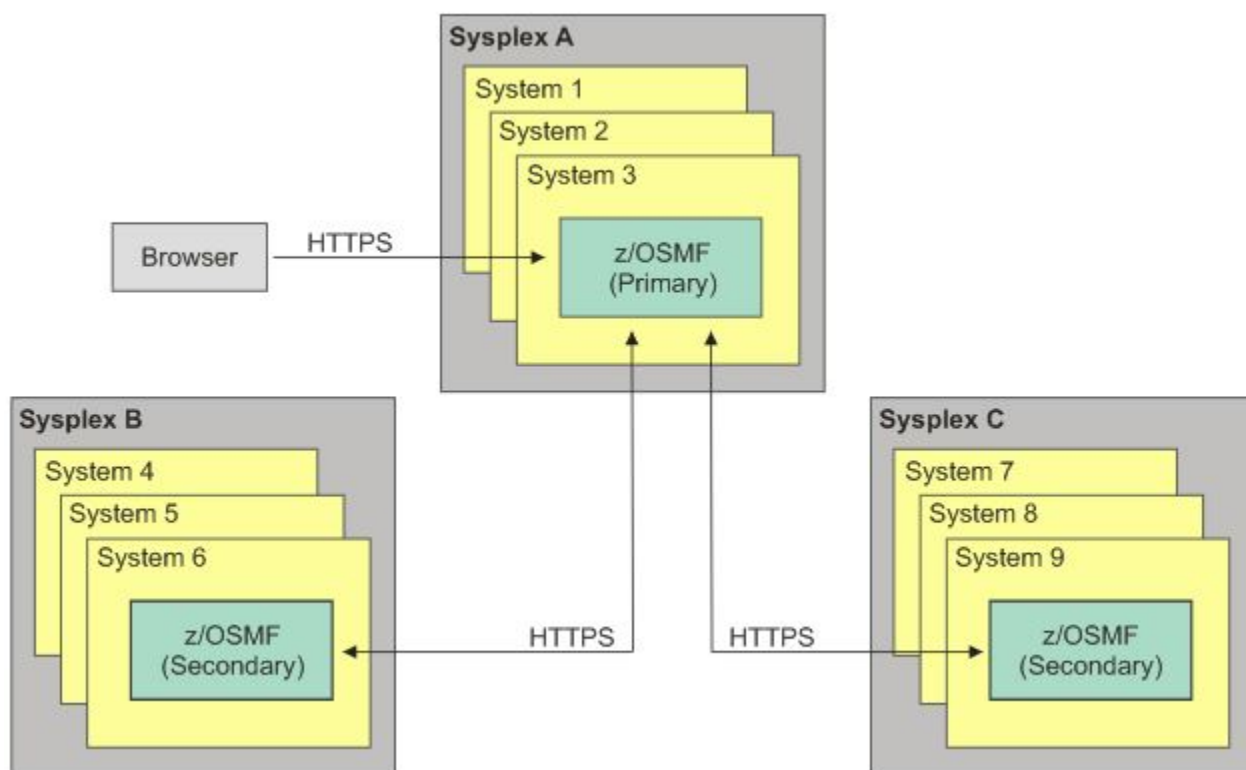


Figure 157. Example sysplex and system configuration

To obtain information from System 6 in Sysplex B, the following actions are performed:

1. A client application sends an HTTPS request to the primary z/OSMF instance.
2. The primary z/OSMF instance routes the HTTPS request to the secondary z/OSMF instance in Sysplex B.
3. The secondary z/OSMF instance processes the request and sends an HTTPS response to the primary z/OSMF instance.
4. The primary z/OSMF instance returns the HTTPS response to the browser.
5. The client application parses the response and extracts the appropriate information.

To route the request, the primary z/OSMF instance needs a system definition that specifies how to access the z/OSMF instance that is running on System 6 in Sysplex B and an HTTP proxy definition that specifies how to navigate the HTTP proxy server that is between the primary and secondary z/OSMF instances.

You can use the z/OSMF Systems task to add or modify the system and HTTP proxy definitions, and you can use the multisystem routing services to route the HTTPS request to the secondary z/OSMF instance and receive the HTTPS response. The remainder of this section describes the multisystem routing services. For information about the Systems task, see the z/OSMF online help.

Multisystem routing services overview

The multisystem routing services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for routing HTTPS requests to multiple systems in your enterprise. The multisystem routing services can route any HTTPS request that is supported by the z/OSMF REST services described in Chapter 1, “Using the z/OSMF REST services,” on page 1.

Table 314 on page 410 lists the operations that the multisystem routing services provide.

Table 314. Operations provided through the multisystem routing services.

Operation	HTTP method and URI path
<u>“Retrieve data from one or more systems” on page 413</u>	GET /zosmf/gateway/system?content=<http-content> GET /zosmf/gateway/systems?content=<http-content> GET /zosmf/gateway/group?content=<http-content> GET /zosmf/gateway/sysplex?content=<http-content> GET /zosmf/gateway/cpc?content=<http-content>
<u>“Update data for one or more systems” on page 420</u>	POST /zosmf/gateway/system POST /zosmf/gateway/systems POST /zosmf/gateway/group POST /zosmf/gateway/sysplex POST /zosmf/gateway/cpc PUT /zosmf/gateway/system PUT /zosmf/gateway/systems PUT /zosmf/gateway/group PUT /zosmf/gateway/sysplex PUT /zosmf/gateway/cpc
<u>“Delete data from one or more systems” on page 426</u>	DELETE /zosmf/gateway/system?content=<http-content> DELETE /zosmf/gateway/systems?content=<http-content> DELETE /zosmf/gateway/group?content=<http-content> DELETE /zosmf/gateway/sysplex?content=<http-content> DELETE /zosmf/gateway/cpc?content=<http-content>
<u>“Authenticate with a secondary z/OSMF instance” on page 432</u>	POST /zosmf/gateway/logon
<u>“Authenticate with an HTTP proxy server” on page 434</u>	POST /zosmf/gateway/logon/proxy

Required authorizations

The user must be logged into z/OSMF. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for HTTP response data. The following JSON object is used by all multisystem routing services for returning data and status about the requested operations. The attributes provided in the JSON object depend on the requested operation.

```
{
  "primaryAPIVersion": "primary-API-version",
  "systemsOutput": {
    "systemOutput": "system-output",
    "rc": "return-code",
    "error": { "msgid": "message-ID", "msgtxt": "message-text" },
    "secondaryApiVersion": "secondary-API-version",
    "systemVersion": "
    {
      "zosNode": "zos-node",
      "zosVrm": "zos-level",
    }
  }
}
```



```

        "zosSysplex": "sysplex-name"
      },
      "systemName": "system-name"
    },
    "numOfSystems": "total-systems"
  }

```

where:

primary-API-version

Version of the multisystem routing services interface for the primary z/OSMF instance.

systemsOutput

Contains a separate response for each system to which the HTTPS request was sent. If the request was sent to multiple systems, the systemsOutput attribute contains an array of system responses.

system-output

Contains the response returned for a single system. A separate systemOutput attribute is included for each system to which the HTTPS request was sent.

return-code

Code returned by the system. The return code can be one of the following values:

OK

Success.

HttpConnectionFailed

The HTTPS connection failed. Typically, this error occurs when the system hosting the secondary z/OSMF instance is unavailable, the z/OSMF instance is not running, or a network error has occurred.

HttpConnectionTimedOut

The HTTPS request did not complete in the time allotted.

CertificateError

The certificate for the secondary z/OSMF instance is not trusted.

LoginRequired

The Lightweight Third Party Authentication (LTPA) token is not valid or has expired. You must submit a separate HTTPS request to authenticate with the z/OSMF instance.

InvalidLogin

The login credentials for the z/OSMF instance are not valid.

ProxyLoginRequired

Authentication is required by the proxy server.

InvalidProxyLogin

The login credentials for the proxy server are not valid.

FailedWithMessage

The request was successful; however, an internal error occurred with the secondary z/OSMF instance.

UnexpectedFailure

An unexpected error occurred.

error

If an error occurred with the request, the error attribute contains the message ID (msgid) and message text (msgtxt) for the message that was issued. Otherwise, this attribute is *null*.

secondary-API-version

Version of the multisystem routing services interface for the secondary z/OSMF instance.

systemVersion

Provides additional information about the system, as follows:

zosNode

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

zosVrm

Version, release, and modification level of the z/OS image installed on the system. The level has the format *vv.rr.mm*, where *vv* is the version, *rr* is the release, and *mm* is the modification level. You can correlate the returned value as follows:

- 04.24.00 indicates z/OS V2R1

zosSysplex

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

system-name

Unique name assigned to the system definition.

total-systems

Number of systems to which an HTTPS request was sent.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate with the primary or secondary z/OSMF instance, or is not authorized to use the z/OSMF REST service.

If the user ID required to authenticate with the primary and secondary z/OSMF instances are not the same, submit a separate HTTPS request to authenticate with the secondary z/OSMF instances.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the multisystem routing services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Retrieve data from one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to retrieve data from one system, from a list of systems, or from all the systems in a group, sysplex, or central processor complex (CPC).

HTTP method and URI path

```
GET /zosmf/gateway/system?content=<http-content>
GET /zosmf/gateway/systems?content=<http-content>
GET /zosmf/gateway/group?content=<http-content>
GET /zosmf/gateway/sysplex?content=<http-content>
GET /zosmf/gateway/cpc?content=<http-content>
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 315 on page 413](#) lists the parameters that are supported for this request.

Important: If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *System#1*, specify *System%231*.

Table 315. Supported input parameters for the multisystem routing services

Parameter	Required	Description
target	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.
resourcePath	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: <code>/tsoApp/ping/<servletKey></code>, where <code><servletKey></code> identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>

Table 315. Supported input parameters for the multisystem routing services (continued)

Parameter	Required	Description
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 410](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 412](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 410](#).

Examples

To obtain sample HTTPS requests and responses for retrieving data from one system, from a list of systems, or from all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Retrieve data from one system” on page 415](#)
- [“Example 2: Retrieve data from a list of systems” on page 415](#)
- [“Example 3: Retrieve data from all the systems in a group” on page 416](#)
- [“Example 4: Retrieve data from all the systems in a sysplex” on page 417](#)
- [“Example 5: Retrieve data from all the systems in a CPC” on page 419](#)

Example 1: Retrieve data from one system

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from system sys057, submit the following request:

```
GET /zosmf/gateway/system?content={\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
{\"contentType\":\"application/json\",\"charset\":\"UTF8\"}} HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 158. Sample request to retrieve data from one system

A sample response is shown in Figure 159 on page 415.

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  \"primaryAPIVersion\":1.0,
  \"systemsOutput\":
  {
    \"systemOutput\":
    {
      \"error\":null,
      \"result\":[
        {
          \"id\":\"IBM.ZOSMF.IZU_IMPORT_HANDLER\",
          \"taskId\":\"IZUG_TASK_zOSMFImportManager\",
          \"enabled\":true,
          \"defaultHandler\":false,
          \"applId\":\"IzuImportManager\",
          \"type\":\"INTERNAL\",
          \"displayName\":\"Import Manager\",
          \"url\":\"/zosmf/IzuImportUtility/index.jsp\",
          \"eventId\":\"IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
          \"options\":{\"CONTEXT_SUPPORT\":\"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH\"}
        }
      ],
      \"rc\":\"Ok\",
      \"secondaryApiVersion\":1.0,
      \"systemVersion\":{\"zosNode\":\"SY1\",\"zosVim\":\"04.24.00\",\"zosSysplex\":\"PLEX1\"},
      \"systemName\":\"sys057\"
    },
    \"numOfSystems\":1
  }
}
```

Figure 159. Sample response from a request to retrieve data from one system

Example 2: Retrieve data from a list of systems

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from system sys057 and for event type IBM.ZOSMF.VIEW_DATASET from system sys060, submit the following request:

```
GET /zosmf/gateway/systems?content=[{\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
{\"contentType\":\"application/json\",\"charset\":\"UTF8\"}},
{\"target\":\"sys060\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.VIEW_DATASET\"}] HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 160. Sample request to retrieve data from a list of systems

A sample response is shown in Figure 161 on page 416.

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventTypeid":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":null,
          "result":[
            {
              "id":"IBM.ISPF.ISR.EPDF.B",
              "enabled":true,
              "defaultHandler":false,
              "type":"EXTERNAL",
              "displayName":"ISR Browse Data Set",
              "url":
                "/zosmf/webispf/index.jsp?cmd=ISPSTART%20CMD(%25ISREPDF%20'%24dataSetName'%20B)
                %20NEWAPPL(ISR)",
              "eventTypeid":"IBM.ZOSMF.VIEW_DATASET",
              "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH"}
            }
          ],
          "rc":"Ok",
          "secondaryApiVersion":1.0,
          "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
          "systemName":"sys060"
        }
      }
    ],
    "numOfSystems":2
  }
}

```

Figure 161. Sample response from a request to retrieve data from a list of systems

Example 3: Retrieve data from all the systems in a group

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in group *mygroup*, submit the following request:

```

GET /zosmf/gateway/group?content={"target":"mygroup",
  "resourcePath":"/izual/rest/handler?eventTypeid=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1

Host: zosmf1.yourco.com

```

Figure 162. Sample request to retrieve data from all the systems in a group

A sample response is shown in [Figure 163 on page 417](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventTypeId":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys058\" failed with return code \"LoginRequired\" and HTTP response code \"401\"."
          },
          "result":null
        },
        "rc":"LoginRequired",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
        "systemName":"sys058"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys059\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
          },
          "result":null
        },
        "rc":"HttpConnectionTimedOut",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
        "systemName":"sys059"
      }
    ],
    "numOfSystems":3
  }
}

```

Figure 163. Sample response from a request to retrieve data from all the systems in a group

Example 4: Retrieve data from all the systems in a sysplex

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in sysplex *PLEX1*, submit the following request:

```
GET /zosmf/gateway/sysplex?content={"target":"PLEX1",
"resourcePath":"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 164. Sample request to retrieve data from all the systems in a sysplex

A sample response is shown in [Figure 165 on page 418](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_ZOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventId":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys077\" failed with return code
              \"LoginRequired\" and HTTP response code \"401\"."
          },
          "result":null
        },
        "rc":"LoginRequired",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys077"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys195\" failed with return code
              \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
          },
          "result":null
        },
        "rc":"HttpConnectionTimedOut",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys195"
      }
    ],
    "numOfSystems":3
  }
}
```

Figure 165. Sample response from a request to retrieve data from all the systems in a sysplex

Example 5: Retrieve data from all the systems in a CPC

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in CPC *CPC1*, submit the following request:

```
GET /zosmf/gateway/cpc?content={"target":"CPC1",  
"resourcePath":"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1  
  
Host: zosmf1.yourco.com
```

Figure 166. Sample request to retrieve data from all the systems in a CPC

A sample response is shown in Figure 167 on page 419.

```
HTTP/1.1 200 OK  
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT  
Connection: close  
  
{  
  "primaryAPIVersion":1.0,  
  "systemsOutput":[  
    {  
      "systemOutput":  
        {  
          "error":null,  
          "result":[  
            {  
              "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",  
              "taskId":"IZUG_TASK_zOSMFImportManager",  
              "enabled":true,  
              "defaultHandler":false,  
              "applId":"IzuImportManager",  
              "type":"INTERNAL",  
              "displayName":"Import Manager",  
              "url":"/zosmf/IzuImportUtility/index.jsp",  
              "eventType": "IBM.ZOSMF.IMPORT_EXTERNAL_APP",  
              "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}  
            }  
          ],  
        },  
      "rc":"Ok",  
      "secondaryApiVersion":1.0,  
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},  
      "systemName":"sys057"  
    },  
    {  
      "systemOutput":  
        {  
          "error":null,  
          "result":null  
        },  
      "rc":"Ok",  
      "secondaryApiVersion":1.0,  
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},  
      "systemName":"sys289"  
    }  
  ],  
  "numOfSystems":2  
}
```

Figure 167. Sample response from a request to retrieve data from all the systems in a CPC

Update data for one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to update data for one system, for a list of systems, or for all the systems in a group, sysplex, or central processor complex (CPC).

HTTP method and URI path

```
POST /zosmf/gateway/system
POST /zosmf/gateway/systems
POST /zosmf/gateway/group
POST /zosmf/gateway/sysplex
POST /zosmf/gateway/cpc
PUT /zosmf/gateway/system
PUT /zosmf/gateway/systems
PUT /zosmf/gateway/group
PUT /zosmf/gateway/sysplex
PUT /zosmf/gateway/cpc
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include a JSON object or JSON object stream that describes the objects to be created, updated, or modified for each system. [Table 316 on page 420](#) lists the supported parameters.

Table 316. Supported input parameters for the multisystem routing services		
Parameter	Required	Description
target	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.

Table 316. Supported input parameters for the multisystem routing services (continued)

Parameter	Required	Description
resourcePath	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: /tsoApp/ping/<servletKey>, where <servletKey> identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 410.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 412.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 410.

Examples

To obtain sample HTTPS requests and responses for updating data for one system, for a list of systems, or for all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Update data for one system”](#) on page 422
- [“Example 2: Update data for a list of systems”](#) on page 422
- [“Example 3: Update data for all the systems in a group”](#) on page 423

- [“Example 4: Update data for all the systems in a sysplex” on page 424](#)
- [“Example 5: Update data for all the systems in a CPC” on page 425](#)

Example 1: Update data for one system

To create event type IBM.ZOSMF.VIEW_JOB_STATUS on system sys057, submit the following request:

```
POST /zosmf/gateway/system HTTP/1.1
Host: zosmf1.yourco.com

{"target":"sys057","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.","owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}}
```

Figure 168. Sample request to update data for one system

A sample response is shown in [Figure 169 on page 422](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "error":null,
      "result":null,
    },
    "rc":"Ok",
    "secondaryApiVersion":1.0,
    "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
    "systemName":"sys057"
  },
  "numOfSystems":1
}
```

Figure 169. Sample response from a request to update data for one system

Example 2: Update data for a list of systems

To create event type IBM.ZOSMF.VIEW_JOB_STATUS on system sys057 and event type IBM.ZOSMF.VIEW_WLM_STATUS on system sys060, submit the following request:

```
POST /zosmf/gateway/systems HTTP/1.1
Host: zosmf1.yourco.com

[{"target":"sys057","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.","owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}},
{"target":"sys060","resourcePath":"/izual/rest/eventtype",
"content":{"id":"IBM.ZOSMF.VIEW_WLM_STATUS","displayName":"View WLM Status",
"desc":"View the status of WLM.","owner":"IBM","params":{"sysplex":"Name of the sysplex."}}}]
```

Figure 170. Sample request to update data for a list of systems

A sample response is shown in [Figure 171 on page 423](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
      "systemName":"sys060"
    }
  ],
  "numOfSystems":2
}

```

Figure 171. Sample response from a request to update data for a list of systems

Example 3: Update data for all the systems in a group

To create event type IBM.ZOSMF.VIEW_JOB_STATUS for all the systems in group *mygroup*, submit the following request:

```

POST /zosmf/gateway/group HTTP/1.1
Host: zosmf1.yourco.com

{"target":"mygroup","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF", "params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 172. Sample request to update data for all the systems in a group

A sample response is shown in [Figure 173 on page 424](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
      "systemName":"sys058"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
      "systemName":"sys059"
    }
  ],
  "numOfSystems":3
}

```

Figure 173. Sample response from a request to update data for all the systems in a group

Example 4: Update data for all the systems in a sysplex

To create event type IBM.ZOSMF.VIEW_JOB_STATUS for all the systems in sysplex *PLEX1*, submit the following request:

```

POST /zosmf/gateway/sysplex HTTP/1.1
Host: zosmf1.yourco.com

{"target":"PLEX1","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF", "params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 174. Sample request to update data for all the systems in a sysplex

A sample response is shown in [Figure 175 on page 425](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys077"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys195"
    }
  ],
  "numOfSystems":3
}

```

Figure 175. Sample response from a request to update data for all the systems in a sysplex

Example 5: Update data for all the systems in a CPC

To create event type IBM.ZOSMF.VIEW_JOB_STATUS for all the systems in CPC *CPC1*, submit the following request:

```

POST /zosmf/gateway/cpc HTTP/1.1
Host: zosmf1.yourco.com

{"target":"CPC1","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 176. Sample request to update data for all the systems in a CPC

A sample response is shown in [Figure 177 on page 426](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},
      "systemName":"sys289"
    }
  ],
  "numOfSystems":2
}

```

Figure 177. Sample response from a request to update data for all the systems in a CPC

Delete data from one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to delete data from one system, from a list of systems, or from all the systems in a group, sysplex, or central processor complex (CPC).

HTTP method and URI path

```

DELETE /zosmf/gateway/system?content=<http-content>
DELETE /zosmf/gateway/systems?content=<http-content>
DELETE /zosmf/gateway/group?content=<http-content>
DELETE /zosmf/gateway/sysplex?content=<http-content>
DELETE /zosmf/gateway/cpc?content=<http-content>

```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 317 on page 427](#) lists the parameters that are supported for this request.

Important: If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *System#1*, specify *System%231*.

Table 317. Supported input parameters for the multisystem routing services

Parameter	Required	Description
target	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.
resourcePath	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: /tsoApp/ping/<servletKey>, where <servletKey> identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>
requestProperties	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
timeout	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
content	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations” on page 410](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 412](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 410](#).

Examples

To obtain sample HTTPS requests and responses for deleting data from one system, from a list of systems, or from all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Delete data from one system” on page 428](#)
- [“Example 2: Delete data from a list of systems” on page 429](#)
- [“Example 3: Delete data from all the systems in a group” on page 429](#)
- [“Example 4: Delete data from all the systems in a sysplex” on page 430](#)
- [“Example 5: Delete data from all the systems in a CPC” on page 431](#)

Example 1: Delete data from one system

To remove handler IBM.ZOSMF.IZU_IMPORT_HANDLER for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from system sys057, submit the following request:

```
DELETE /zosmf/gateway/system?content={\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\"}
HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 178. Sample request to delete data from one system

A sample response is shown in [Figure 179 on page 428](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  \"primaryAPIVersion\":1.0,
  \"systemsOutput\":
    {
      \"systemOutput\":
        {
          \"error\":null,
          \"result\":null
        },
      \"rc\":\"Ok\",
      \"secondaryApiVersion\":1.0,
      \"systemVersion\":{\"zosNode\":\"SY1\", \"zosVim\":\"04.24.00\", \"zosSysplex\":\"PLEX1\"},
      \"systemName\":\"sys057\"
    },
  \"numOfSystems\":1
}
```

Figure 179. Sample response from a request to delete data from one system

Example 2: Delete data from a list of systems

To remove handler IBM.ZOSMF.IZU_IMPORT_HANDLER for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from system sys057 and to remove event type IBM.ZOSMF.VIEW_DATASET from system sys060, submit the following request:

```
DELETE /zosmf/gateway/systems?content=[{"target":"sys057",
"resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"},
{"target":"sys060","resourcePath":"/izual/rest/eventtype/IBM.ZOSMF.VIEW_DATASET"}] HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 180. Sample request to delete data from a list of systems

A sample response is shown in [Figure 181](#) on page 429.

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG698E",
          "msgtxt":"The request could not be completed because 1 handlers are registered for
event type \"IBM.ZOSMF.VIEW_DATASET\"."
        },
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
      "systemName":"sys060"
    }
  ],
  "numOfSystems":2
}
```

Figure 181. Sample response from a request to delete data from a list of systems

Example 3: Delete data from all the systems in a group

To remove handler IBM.ZOSMF.IZU_IMPORT_HANDLER for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in group *mygroup*, submit the following request:

```
DELETE /zosmf/gateway/group?content={"target":"mygroup",
"resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 182. Sample request to delete data from all the systems in a group

A sample response is shown in [Figure 183 on page 430](#).

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
      "systemName":"sys058"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG0000E",
          "msgtxt":"The HTTPS request to server \"sys059\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
        },
        "result":null
      },
      "rc":"HttpConnectionTimedOut",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
      "systemName":"sys059"
    }
  ],
  "numOfSystems":3
}
```

Figure 183. Sample response from a request to delete data from all the systems in a group

Example 4: Delete data from all the systems in a sysplex

To remove handler IBM.ZOSMF.IZU_IMPORT_HANDLER for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in sysplex *PLEX1*, submit the following request:

```
DELETE /zosmf/gateway/sysplex?content={"target":"PLEX1",
"resourcePath":"/izual/test/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 184. Sample request to delete data from all the systems in a sysplex

A sample response is shown in [Figure 185 on page 431](#).

```

HTTP/1.1 200 OK
Date: Fri, 16 Feb 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys077"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG0000E",
          "msgtxt":"The HTTPS request to server \"sys195\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
        },
        "result":null
      },
      "rc":"HttpConnectionTimedOut",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys195"
    }
  ],
  "numOfSystems":3
}

```

Figure 185. Sample response from a request to delete data from all the systems in a sysplex

Example 5: Delete data from all the systems in a CPC

To remove handler IBM.ZOSMF.IZU_IMPORT_HANDLER for event type IBM.ZOSMF.IMPORT_EXTERNAL_APP from all the systems in CPC *CPC1*, submit the following request:

```

DELETE /zosmf/gateway/cpc?content={"target":"CPC1",
  "resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
  eventType=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com

```

Figure 186. Sample request to delete data from all the systems in a CPC

A sample response is shown in [Figure 187 on page 432](#).

```

HTTP/1.1 200 OK
Date: Fri, 16 Feb 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},
      "systemName":"sys289"
    }
  ],
  "numOfSystems":2
}

```

Figure 187. Sample response from a request to delete data from all the systems in a CPC

Authenticate with a secondary z/OSMF instance

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to authenticate with a secondary z/OSMF instance.

HTTP method and URI path

```
POST /zosmf/gateway/logon
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **logon** informs the service that the request is to authenticate with a system.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include the following JSON object:

```

{
  "userid":"user-ID",
  "password":"password",

```

```
"systemName": "system-name"
}
```

where:

user-ID

z/OS user ID that allows the user to access the specified system. The user ID is the same user ID that is specified in your installation's z/OS security management facility (for example, RACF). The user ID is required.

password

Password or pass phrase associated with the z/OS user ID. The password is required.

system-name

Unique name assigned to the system definition.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 410](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 412](#).

The response also includes a JSON object that indicates whether the request was successful. If the logon request is successful, the timeout for the Lightweight Third Party Authentication (LTPA) token is returned, as depicted in [Figure 188 on page 433](#).

```
{"timeout":7564710}
```

Figure 188. Successful response when authenticating with a system

If the logon request is unsuccessful, the JSON object contains an error message, as depicted in [Figure 189 on page 433](#).

```
{"error":true,"errMsg":"IZUG410E: The user ID, password, or pass phrase is not valid.  
Enter the correct values for your security management product."}
```

Figure 189. Response when the authentication request fails

Example

To authenticate with system `sys057`, submit the following request:

```
POST /zosmf/gateway/logon HTTP/1.1
Host: zosmf1.yourco.com

{"userid":"claire","password":"abc123","systemName":"sys057"}
```

Figure 190. Sample request to authenticate with a system

Authenticate with an HTTP proxy server

You can use this operation to authenticate with the HTTP proxy server that the primary z/OSMF instance is required to navigate to communicate with a secondary z/OSMF instance.

HTTP method and URI path

```
POST /zosmf/gateway/logon/proxy
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **logon/proxy** informs the service that the request is to authenticate with an HTTP proxy server.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include the following JSON object:

```
{
  "proxyUserId": "proxy-user-ID",
  "proxyPassword": "proxy-password",
  "systemName": "system-name"
}
```

where:

proxy-user-ID

User ID that allows the user to access the HTTP proxy server at your enterprise. The user ID is required.

proxy-password

Password or pass phrase associated with the proxy user ID. The password is required.

system-name

Unique name assigned to the system definition that specifies the URL for accessing the secondary z/OSMF instance.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 410](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 412](#).

The response also includes a JSON object that indicates whether the request was successful. If the logon request is successful, *null* values are returned for the *result* and *error* attributes, as depicted in [Figure 191](#) on page 435.

```
{"result":null,"error":null}
```

Figure 191. Successful response when authenticating with an HTTP proxy server

If the logon request is unsuccessful, the JSON object contains an error message, as depicted in [Figure 192](#) on page 435. For a description of each attribute, see “Content type used for HTTP response data” on page 410.

```
{
  "primaryAPIVersion":1.0,
  "systemsOutput":null,
  "error":{
    "msgid":"IZUG476E",
    "msgtxt":"The HTTP request to the secondary z/OSMF instance \"sys057\"
    failed with error type \"InvalidProxyLogin\" and response code \"407\"."
  },
  "numOfSystems":0
}
```

Figure 192. Sample response when the authentication request fails

Example

To authenticate with the HTTP proxy server that is between the primary z/OSMF instance and the z/OSMF instance that is running on system *sys057*, submit the following request:

```
POST /zosmf/gateway/logon/proxy HTTP/1.1
Host: zosmf1.yourco.com

{"proxyUserId":"claire","proxyPassword":"abc123","systemName":"sys057"}
```

Figure 193. Sample request to authenticate with an HTTP proxy server

MVS subsystem services

The MVS subsystem services API is provided for z/OSMF tasks and vendor applications. This API is used to list the MVS subsystems on a z/OS system.

Table 318. Subsystem services method	
Operation	HTTP method and URI path
“List MVS subsystems” on page 436	GET /zosmf/rest/mvssubs

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 206 Partial content

Request was processed successfully, however, only a portion of the available content was received. The request contained the X-IBM-Max-Items header, which limited the amount of content that was returned.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 405 Method not allowed

Requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

HTTP 503 Service unavailable

Server is not available.

Error logging

Errors from the MVS subsystem services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

List MVS subsystems

You can use the GET method to list the subsystems on a z/OS system. You can filter the returned list of subsystems by specifying a subsystem id or wild-card.

HTTP method and URI path

```
GET /zosmf/rest/mvssubs
GET /zosmf/rest/mvssubs?ssid=filter-criteria
```

Where:

- **/zosmf/rest** specifies the z/OSMF REST services API.
- **/mvssubs** indicates an MVS subsystems request.

Query parameters**ssid**

An optional query parameter that can be used to qualify the request.

Response Body

If the request is successfully executed, will return 200 status code. In all cases an application/json document will be returned:

Table 319. Response

Property	Description	Required
subsys	Subsystem name. The subsystem ID or hexadecimal ID if the subsystem ID is not printable.	Yes
active	True if the subsystem is active, otherwise false.	Yes
funcs	An array of integer values, representing the subsystem function IDs that are defined by this subsystem.	Yes
primary	True if the subsystem is the primary subsystem.	No
dynamic	True if the subsystem is a dynamically defined subsystem.	No
commands	True for a dynamic subsystem that supports commands.	No
incomplete	True if the returned information for a dynamic subsystem is incomplete.	No
eventrtn	True if a dynamic subsystem has an event routine.	No
JSONversion	JSON version.	No

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example

Refer to [Figure 194 on page 438](#) for an example of a list of MVS subsystems.

```
request:GET https://zosmf1.yourco.com/zosmf/rest/mvssubs HTTP/1.1
Content-type: application/json; charset=UTF-8
JSON response document:

{"items":[
  {"subsys":"JES2", "active":true, "primary":true, "dynamic":true,
    "funcs":[1,2,3,4,5,6,7,8,9,10,11,12,13,16,17,18,19,20,21,53,54,64,70,71,75,77,79,80,81,82,83,84,85]},
  {"subsys":"MSTR", "active":true, "funcs":[4,5,6,8,9,10,12,14,15,32,33,48,50,54,63,68,72,73,78,80]},
  {"subsys":"SMS ", "active":true, "dynamic":true, "funcs":[8,15,16,17,55]},
  {"subsys":"RACF", "active":true, "dynamic":true, "funcs":[10,16,17,39,255]},
  {"subsys":"IRLM", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"JRLM", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"LOGR", "active":true, "dynamic":true, "funcs":[7,16,17,18,19,38,39]},
  {"subsys":"RRS ", "active":true, "dynamic":true, "funcs":[10,54]},
  {"subsys":"RRSA", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"BLSR", "active":true, "dynamic":true, "funcs":[7,16,17,38,39]},
  {"subsys":"ISPF", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"DSN9", "active":true, "funcs":[4,8,10,41,50]},
  {"subsys":"IRL9", "active":true, "funcs":[51]},
  {"subsys":"AXR ", "active":true, "dynamic":true, "funcs":[10]},
  {"subsys":"03E20023", "active":true, "funcs":[4,8,10]}
],
"JSONversion":1}
```

Figure 194. List MVS subsystems

Notification services

The Notification services are provided for z/OSMF tasks and vendor applications. These services are used to send a notification in the form of a notification record or email, to a single or multiple recipients. On a successful request, all of the recipients get the notification in their z/OSMF Notification task as the default destination.

Table 320. Notification methods

Operation	HTTP method and URI path
<u>“Get all of the notifications received by the current user” on page 439</u>	GET /zosmf/notifications/inbox
<u>“Send a notification from a z/OSMF task, when the content is the message from the bundle file ” on page 441</u>	POST /zosmf/notifications/new
<u>“Send a notification and mail from a z/OSMF task or z/OSMF user” on page 443</u>	POST /zosmf/notifications/new

Using the Swagger interface

You can use the Swagger interface to display information about the Notification services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 893](#).

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 404 Not found

Requested resource does not exist.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

Error logging

Errors from the z/OSMF notifications services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Get all of the notifications received by the current user

You can use this operation to get all of the notifications that were received by the current user. This operation supports only the user to get notification items in the z/OSMF Notifications task. This does not apply to the get mail operation in a user's email account.

HTTP method and URI path

```
GET /zosmf/notifications/inbox
```

Query Parameters

None.

Request

None.

Response Content

On successful completion, the service returns a response body, which contains details about the notifications. [Table 321 on page 439](#) lists the fields in the response body.

Table 321. Response content from the notifications received by the current user		
Field	Type	Description
taskId	String	This is the ID of the task where the notification is initiated.
pluginId	String	This is the ID of the plug-in where the notification is initiated.
appLinkEventId	String	This is the event ID for an application event.
assignees	String	The user IDs, group names, or email addresses of all the recipients. These values are represented as a string, and are separated by a comma.

Table 321. Response content from the notifications received by the current user (continued)

Field	Type	Description
descriptionParms	String	These are the parameters which substitute the description.
defaultDescription	String	This is the default description of the notification.
descriptionId	String	This is the message ID of the description.
descriptionBundleURL	String	This is the bundle file where the description of the notification can be found.
appLinkParms	String	This is the map of the parameters, which are sent with an event.

Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF. See [“Authenticating to z/OSMF”](#) on page 3.

You will also need SAF authority, as described in [Security structures for z/OSMF](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided, see [Table 321](#) on page 439.

The HTTP status codes and error handling are described in [“Error handling”](#) on page 438.

Example

Receive messages in your inbox.

Request

```
GET /zosmf/notifications/inbox
```

Response

```
{
  "items": [
    {
      "appLinkHandlers": true,
      "taskId": "Workflows",
      "pluginID": "workflow",
      "descriptionBundle": "WorkflowMessages",
      "appLinkEventID": "IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW",
      "userRead": false,
      "assignees": "zosmfad",
      "descriptionParms": ["testing -service"],
      "defaultDescription":
        "One or more steps in workflow \"testing -service\" have been assigned to
you.",
      "descriptionID": "IZUWF0040I",
      "timestamp": "1429860854757",
      "notificationID": "1429860854756",
      "bundleUrl": "\/zosmf\/workflow\/js\/zosmf",
      "descriptionBundleURL": "\/zosmf\/workflow\/js\/zosmf\/",
      "defaultTaskName": "Workflows",
      "appLinkParms": {
        "workflow_name": "testing -service"
      },
      "bundleName": "taskBundle",
      "numUnreadNotification": 1
    }
  ]
}
```

Send a notification from a z/OSMF task, when the content is the message from the bundle file

This operation is used to send a notification from a z/OSMF task and the content of the notification is the message from the bundle file.

HTTP method and URI path

POST /zosmf/notifications/new

Query Parameters

None

Description

The content of the notification contains the message from the bundle file. This operation supports application linking from the notification task to the receiver task. The destination of a notification will depend on user preferences.

Request Content

A notification is sent from a z/OSMF task. The content should include the message ID and the message text, which originate from a bundle file. In this case the value of the post body should be JSON Object-like. See [Table 322 on page 441](#).

Table 322. Request content for the send notification request			
Input	Description	Type	Required or Optional
pluginId	ID of the plug-in where the notification is initiated.	String	Required
taskId	ID of the task where the notification is initiated.	String	Required
assignees	The user IDs, group names, or email addresses of all the recipients.	String	Required
sendTo	The assignees will receive the notification in the form of email to their configured email address. Mail is the only supported value of this parameter.	String	Required
descriptionBundleURL	Bundle file where the description of the notification can be found.	String	Required
descriptionId	Message ID of the description.	String	Required

Table 322. Request content for the send notification request (continued)			
Input	Description	Type	Required or Optional
defaultDescription	Default description of the notification.	String	Required
descriptionParms	Parameters that substitute the description.	String	Optional
appLinkEventId	Event ID for an application event.	String	Optional
appLinkParms	Map of the parameters, which are sent with an event.	String	Optional

Response Content

On completion, the request returns a JSON object with details about the notification. The response content is shown in [Table 323 on page 442](#).

Table 323. Response content for the send notification request		
Field	Type	Description
apiVersion	String	The version of the Notification Services API.
result	JSONObject	Contains all of the output for each notification destination. It includes 1-3 keys depending on how many destinations the notification is sent to. Each key represents one destination, and its value is a JSONObject which might have messages and return codes.

Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF. See [“Authenticating to z/OSMF” on page 3](#).

You will also need SAF authority, as described in [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided, [Table 323 on page 442](#).

The HTTP status codes and error handling are described in [“Error handling” on page 438](#).

Escaping special characters

The format of a notification should be a valid JSONObject. If a special character exists it must be escaped. If " exists, it needs to be escaped as \". If \ exists, it needs to be escaped as \\.

Example

The notification is only sent to assignees' email inbox

Request

```
POST /zosmf/notifications/new
{"subject": "Test Notification Framework",
 "content": "This is a test.",
 "assignees": "user1@abc.com,zosmfad,user2@abc.com",
 "sendTo": "mail"}
```

Response

```
{"result":{"mail":{"rc":"Ok","messages":{}}},"apiVersion":"1.0"}
```

Send a notification to user ID's and a group.

Request

```
POST /zosmf/notifications/new
{  "pluginId": "workflow",
   "taskId": "Workflows",
   "assignees": "zmfuser, zosmfad, z/OSMF Administrators",
   "descriptionBundleURL": "/zosmf/workflow/js/zosmf/",
   "descriptionId": "IZUWF0039I",
   "defaultDescription": "This is a default description.",
   "applLinkId": "IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW",
   "applLinkParams": {"workflow_name": "new workflow"}}
data:{
  "event": {
    "dte": "15/12/31",
    "tme": "04:29:57",
    "sys": "SYS1",
    "cat": "2",
    "col": "red",
    "msg": "event 1 (of event list)",
    "lng": "long message",
    "viw": {"workflows":{"key":"13309779173140.992077"}}
  }
}
```

Response

```
{"apiVersion":"1.0",
 "result":{"mail":{"messages":null,
 "rc":"Ok"},
 "notification":{"messages":null}}}
}
```

Send a notification and mail from a z/OSMF task or z/OSMF user

This operation is used to send a notification from a z/OSMF task or a z/OSMF user.

HTTP method and URI path

```
POST /zosmf/notifications/new
```

Query Parameters

None

Description

The content of the notification as well as the mail contains the user input data. This operation does not support application linking. A notification with the same subject and content will be sent to all recipients. If the "attachment" parameter is specified, the attachment will only appear in the recipients' mail.

Request Content

The value of the post body should be JSON Object-like. See [Table 324 on page 444](#).

<i>Table 324. Request content from a notification that requires user input</i>			
Input	Description	Type	Required or Optional
assignees	The user IDs, group names, or email addresses of all the recipients.	String	Required
subject	Subject of the notification. The allowable length is 1-500.	String	Required
content	Notification body. The allowable length is 0-5000.	String	Optional
sendTo	The assignees will receive the notification in the form of email to their configured email address. Mail is the only supported value of this parameter.	String	Required
attachment	Array of the file paths, up to 5 attachments are allowed.	String	Optional
pluginId	ID of the plug-in where the notification is initiated.	String	Required, only if the notification is from a z/OSMF task.
taskId	ID of the task where the notification is initiated.	String	Required, only if the notification is from a z/OSMF task.

Response Content

On completion, the request returns a JSON object with details about the notification. The response content is shown in [Table 325 on page 444](#).

<i>Table 325. Response content from a notification that requires user input</i>		
Field	Type	Description
apiVersion	String	The version of the Notification Services API.

Table 325. Response content from a notification that requires user input (continued)

Field	Type	Description
result	JSONObject	Contains all of the output for each notification destination. It includes 1-3 keys depending on how many destinations the notification is sent to. Each key represents one destination, and its value is a JSONObject which might have messages and return codes.

Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

You will also need SAF authority, as described in [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided. See [Table 325 on page 444](#).

The HTTP status codes and error handling are described in [“Error handling” on page 438](#).

Escaping special characters

The format of a notification should be a valid JSONObject. If a special character exists it must be escaped. If " exists, it needs to be escaped as \". If \ exists, it needs to be escaped as \\.

Example

The notification is only sent to assignees' email inbox

Request

```
POST /zosmf/notifications/new
{"subject":"Test Notification Framework",
 "content":"This is a test.",
 "assignees":"user1@abc.com,zosmfad,user2@abc.com",
 "sendTo":"mail"}
```

Response

```
{"result":{"mail":{"rc":"Ok","messages":{}}},"apiVersion":"1.0"}
```

Send a notification with an attachment to a user ID.

Request

```
POST zosmf/notifications/new
{"subject":"Test with unix attachment",
 "content":"See if there is an attachment.",
 "assignees":"zosmfad",
 "attachment":["\\/global/zosmf/data/logs/IZUG0.log\\"]}
```

Response

```
{"apiVersion":"1.0",
 "result":{"mail":{"messages":{"errorData":[{"messageText":"IZUG615E: The
```

```

connection to the SMTP host \"smtp.gmail.com\" port \"587\" failed with
error type \"ConnectionTimedout\".",
"messageId": "IZUG615E"}]},
"rc": "ConnectionTimedout"}},
"notification": {"messages": null}}

```

Security Configuration Assistant services

The Security Configuration Assistant (SCA) REST interface is an application programming interface, which is implemented through industry standard Representational State Transfer (REST) services.

You can use the SCA REST interface to validate security requirements by specifying the JSON format of security requirements either in the HTTP request body, or alternatively, in a static JSON security descriptor file. This new REST service allows you to focus on security requirements that are organized by product or function, instead of the need to understand complex command syntax and differences among various security products.

For more information on how to configure SCA REST interface, see [Configure the Security Configuration Assistant service in IBM z/OS Management Facility Configuration Guide](#).

Table 326 on page 446 lists the operations that the SCA services provide.

Table 326. Operations provided through the SCA services.	
Operation	HTTP method and URI path
“Validate security resources defined in an HTTP request body” on page 447	POST /zosmf/config/security/version/validate?userid=<user-id>
“Validate security resources defined in a security descriptor file” on page 451	POST /zosmf/config/security/version/validate/descriptor?userid=<user-id>
“Provision security resources defined in an HTTP request body” on page 455	POST /zosmf/config/security/version/provision?userid=<user-id>

Required authorizations

The user must be logged in to z/OSMF, and must have READ access to the SAF profile that was registered for the plug-in and task making the request.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains the following attributes:

```
{
  "messageId": "message-id",
  "messageText": "message-text "
}
```

The following HTTP status codes are valid:

HTTP 400

The request contained a parameter error or is missing.

HTTP 403

User is not logged on or is not allowed to access z/OSMF Security Configuration Assistant.

HTTP 500

Internal Server Error.

Error logging

Errors from the security configuration assistant services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Validate security resources defined in an HTTP request body

Use this operation to validate security resources defined in an HTTP request body.

HTTP method and URI path

```
POST /zosmf/config/security/<version>/validate?userid=<user-id>
```

where:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid: v1.

Query parameter

userid

Identifies the user ID or group ID in which the security resources specified in the request body will be validated for.

If the user ID is not specified, the current logon user ID will be used to validate the security resources in the security descriptor file.

Request content

Your request must include a JSON object that describes the security resources to be validated. For additional details, see [Table 327 on page 448](#).

Table 327. Request content to validate security resources in a post body

Field name	Alias name	Type	Required or optional	Description
serviceId	ServiceId	String	Optional	Indicates the service ID.
serviceName	ServiceName	String	Optional	Indicates the service name.
version	MetaValidationItemVersion	Number	Optional	Indicates the version.
vendor	Vendor	String	Optional	Indicates the vendor name.
resourceItems	SecurityValidationItems	Object	Required	Indicates the security resources to be validated for the specified user ID.
itemId	ItemId, ItemID	String	Optional	Indicates the item ID.
itemType	ItemType	String	Optional	Indicates the item type. <ul style="list-style-type: none"> • If the value is not specified for this property, the current resource will be used for validation. • If the value specified for this property resource indicates any value other than 'PROGRAMMABLE', it will be ignored and not used for validation. The resource will not be validated and will not appear in the response list of the validated resources.
itemCategory	ItemCategory	String	Optional	Indicates the item category.
itemDescription	ItemDescription	String	Optional	Indicates the item description.
resourceProfile	ResourceProfile	String	Required	Indicates the name of the security resource profile. A generic resource name and a variable in the name is not supported.
resourceClass	ResourceClass	String	Required	Indicates the SAF resource class.
access	LevelOfAccessRequired	String	Required	Indicates the level of access that is required to the security resource for the specified user ID or group ID. Possible values include: <ul style="list-style-type: none"> • READ • UPDATE • CONTROL • ALTER

Table 327. Request content to validate security resources in a post body (continued)

Field name	Alias name	Type	Required or optional	Description
whoNeedsAccess	WhoNeeds Access	String	Optional	Indicates the users (security groups) who require access to this resource. The Security Configuration Assistant does not verify that security groups are defined; your security administrator must verify that the groups exist.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

Response content

On successful completion, HTTP status code 200 (OK) is returned. The response also includes a JSON object that contains the requested information. For additional details, see [Table 328 on page 449](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message. For additional details, see [“Error handling” on page 447](#).

Table 328. Response content to validate security resources in a post body.

Field name	Type	Required or optional	Description
serviceId	String	Optional	Indicates the service ID.
serviceName	String	Optional	Indicates the service name.
version	Number	Optional	Indicates the version.
vendor	String	Optional	Indicates the vendor name.
resourceItems	Object	Required	Indicates the security resources to be validated for the specified user ID.
itemId	String	Optional	Indicates the item ID.
itemType	String	Optional	Indicates the item type. <ul style="list-style-type: none"> If the value is not specified for this property, the current resource will be used for validation. If the value specified for this property resource indicates any value other than ‘PROGRAMMABLE’, it will be ignored and not used for validation. The resource will not be validated and will not appear in the response list of the validated resources.
itemCategory	String	Optional	Indicates the item category.
itemDescription	String	Optional	Indicates the item description.
resourceProfile	String	Required	Indicates the name of the security resource profile. A generic resource name and a variable in the name is not supported.
resourceClass	String	Required	Indicates the SAF resource class.

Table 328. Response content to validate security resources in a post body. (continued)

Field name	Type	Required or optional	Description
access	String	Required	Indicates the level of access that is required to the security resource for the specified user ID or group ID. Possible values include: <ul style="list-style-type: none"> • READ • UPDATE • CONTROL • ALTER
action	String	Required	For the validation action, the return value will be “validate”.
actionObjectId	String	Required	Indicates the object ID of this action. For the validation action, this ID is the same as <i>validatedId</i> below. This field can also be used for other actions in future versions.
validatedId	String	Required	Indicates the user ID or group ID that is used to validate the resource.
status	String	Required	Indicates the validation result. Possible values include: <ul style="list-style-type: none"> • passed • failed • unknown
additionalInfo	String	Optional	Indicates additional information.
whoNeedsAccess	String	Optional	Indicates the users (security groups) who require access to this resource. The Security Configuration Assistant does not verify that security groups are defined; your security administrator must verify that the groups exist.
messageId	String	Optional	Indicates the message ID.
messageText	String	Optional	Indicates the message text.
httpStatus	Integer	Optional	Indicates the HTTP status code if an error is encountered.
requestMethod	String	Optional	Indicates the HTTP request method if an error is encountered.
requestUri	String	Optional	Indicates the URI if an error is encountered.

Example

In the following example, the POST method is used to validate security resources.


```
POST /zosmf/config/security/v1/validate?userid=ibmtest
{
  "resourceItems": [{
    "resourceProfile": "IZUDFLT.ZOSMF2",
    "resourceClass": "ZMFAPLA",
    "access": "READ"
  }, {
    "resourceProfile": "IZUDFLT.ZOSMF3",
    "resourceClass": "ZMFAPLA",
    "access": "READ"
  }]
}
```

Figure 195. Sample request body to validate security resources.

Figure 196 on page 451 provides a sample response from the validate security resources request.

```
{
  "resourceItems": [{
    "resourceProfile": "IZUDFLT.ZOSMF2",
    "resourceClass": "ZMFAPLA",
    "access": "READ",
    "action": "validate",
    "actionObjectId": "ibmtest",
    "validatedId": "ibmtest",
    "status": "passed"
  }, {
    "resourceProfile": "IZUDFLT.ZOSMF3",
    "resourceClass": "ZMFAPLA",
    "access": "READ",
    "action": "validate",
    "actionObjectId": "ibmtest",
    "validatedId": "ibmtest",
    "status": "unknown",
    "additionalInfo": "Validation returns unknown because ..."
  }]
}
```

Figure 196. Sample response body from the validate security resources request.

Validate security resources defined in a security descriptor file

Use this operation to validate security resources defined in a security descriptor file.

HTTP method and URI path

```
POST /zosmf/config/security/<version>/validate/descriptor?userid=<user-id>
```

where:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid: v1.

Query parameter

userid

Identifies the user ID or group ID in which the security resources specified in the security descriptor file will be validated for.

If the user ID is not specified, the current logon user ID will be used to validate the security resources in the security descriptor file.

Request content

Your request must include a JSON object that describes the security resources to be validated. For additional details, see [Table 329 on page 452](#) and [Table 330 on page 452](#).

Table 329. Request content used to validate security resources in a descriptor file.

Field name	Type	Required or optional	Description
path	String	Required	Absolute path of the existing security descriptor JSON file, which contains resources to be validated against the specified user ID query parameter. The current logged-on user ID is used to read the specified JSON file. If the current logged-on user does not have proper access to the JSON file, an error is returned.

Table 330. JSON security descriptor fields as indicated by the path field.

Field name	Alias name	Type	Required or optional	Description
serviceId	ServiceId	String	Optional	Indicates the service ID.
serviceName	ServiceName	String	Optional	Indicates the service name.
version	MetaValidationItemVersion	Number	Optional	Indicates the version.
vendor	Vendor	String	Optional	Indicates the vendor name.
resourceItems	SecurityValidationItems	Object	Required	Indicates the security resources to be validated for the specified user ID.
itemId	ItemId, ItemID	String	Optional	Indicates the item ID.
itemType	ItemType	String	Optional	Indicates the item type. <ul style="list-style-type: none"> If the value is not specified for this property, the current resource will be used for validation. If the value specified for this property resource indicates any value other than 'PROGRAMMABLE', it will be ignored and not used for validation. The resource will not be validated and will not appear in the response list of the validated resources.
itemCategory	ItemCategory	String	Optional	Indicates the item category.
itemDescription	ItemDescription	String	Optional	Indicates the item description.
resourceProfile	ResourceProfile	String	Required	Indicates the name of the security resource profile. A generic resource name and a variable in the name is not supported.
resourceClass	ResourceClass	String	Required	Indicates the SAF resource class.

Table 330. JSON security descriptor fields as indicated by the path field. (continued)

Field name	Alias name	Type	Required or optional	Description
access	LevelOfAccessRequired	String	Required	Indicates the level of access that is required to the security resource for the specified user ID or group ID. Possible values include: <ul style="list-style-type: none"> • READ • UPDATE • CONTROL • ALTER
whoNeedsAccess	WhoNeedsAccess	String	Optional	Indicates the users (security groups) who require access to this resource. The Security Configuration Assistant does not verify that security groups are defined; your security administrator must verify that the groups exist.

Authorization requirements

The user must be a provisioning administrator or a domain administrator.

For more information, see [“Resource management services” on page 102](#).

Response content

On successful completion, HTTP status code 200 (OK) is returned. The response also includes a JSON object that contains the requested information. For additional details, see [Table 331 on page 453](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message. For additional details, see [“Error handling” on page 447](#).

Table 331. Response content to validate security resources in a security descriptor file.

Field name	Type	Required or optional	Description
serviceId	String	Optional	Indicates the service ID.
serviceName	String	Optional	Indicates the service name.
version	Number	Optional	Indicates the version.
vendor	String	Optional	Indicates the vendor name.
resourceItems	Object	Required	Indicates the security resources to be validated for the specified user ID.
itemId	String	Optional	Indicates the item ID.

Table 331. Response content to validate security resources in a security descriptor file. (continued)

Field name	Type	Required or optional	Description
itemType	String	Optional	Indicates the item type. <ul style="list-style-type: none"> • If the value is not specified for this property, the current resource will be used for validation. • If the value specified for this property resource indicates any value other than 'PROGRAMMABLE', it will be ignored and not used for validation. The resource will not be validated and will not appear in the response list of the validated resources.
itemCategory	String	Optional	Indicates the item category.
itemDescription	String	Optional	Indicates the item description.
resourceProfile	String	Required	Indicates the name of the security resource profile. A generic resource name and a variable in the name is not supported.
resourceClass	String	Required	Indicates the SAF resource class.
access	String	Required	Indicates the level of access that is required to the security resource for the specified user ID or group ID. Possible values include: <ul style="list-style-type: none"> • READ • UPDATE • CONTROL • ALTER
action	String	Required	For the validation action, the return value will be "validate".
actionObjectId	String	Required	Indicates the object ID of this action. For the validation action, this ID is the same as <i>validatedId</i> below. This field can also be used for other actions in future versions.
validatedId	String	Required	Indicates the user ID or group ID that is used to validate the resource.
status	String	Required	Indicates the validation result. Possible values include: <ul style="list-style-type: none"> • passed • failed • unknown
additionalInfo	String	Optional	Indicates additional information.
whoNeedsAccess	String	Optional	Indicates the users (security groups) who require access to this resource. The Security Configuration Assistant does not verify that security groups are defined; your security administrator must verify that the groups exist.
messageId	String	Optional	Indicates the message ID.
messageText	String	Optional	Indicates the message text.

Table 331. Response content to validate security resources in a security descriptor file. (continued)

Field name	Type	Required or optional	Description
httpStatus	Integer	Optional	Indicates the HTTP status code if an error is encountered.
requestMethod	String	Optional	Indicates the HTTP request method if an error is encountered.
requestUri	String	Optional	Indicates the URI if an error is encountered.

Example

In the following example, the POST method is used to validate security resources.

```
POST /zosmf/config/security/v1/validate/descriptor?userid=ibmtest
{
  "path" : "/usr/lpp/zosmf/configuration/izu5655S280100.json"
}
```

Figure 197. Sample request body to validate security resources.

Figure 198 on page 455 provides a sample response from the validate security resources request.

```
{
  "serviceId": "5655S280100",
  "serviceName": "z/OSMF ISPF",
  "version": "1.0",
  "vendor": "IBM",
  "resourceItems": [{
    "itemID": "5655S28TS00I00003000",
    "itemType": "PROGRAMMABLE",
    "itemCategory": "TSO/E Address Space Services",
    "resourceProfile": "CEA.CEATSO.TSOREQUEST",
    "resourceClass": "SERVAUTH",
    "whoNeedsAccess": "<User of the Service>",
    "access": "READ",
    "itemDescription": "Allows the user to use common event adapter (CEA).",
    "action": "validate",
    "actionObjectId": "ibmtest",
    "validatedId": "ibmtest",
    "status": "unknown",
    "additionalInfo": "resources defined in JSON file but not validated due to variables in name or non-programmable type will not appear in the response list"
  }
]
```

Figure 198. Sample response body from the validate security resources request.

Provision security resources defined in an HTTP request body

Use this operation to provision security resources defined in an HTTP request body.

HTTP method and URI path

```
POST /zosmf/config/security/<version>/provision?userid=<user-id>
```

where:

<version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:
v1.

Query parameter

userid

Identifies the user ID or group ID for which the security updates in descriptor will be authorized.

If the user ID or group ID is not specified, the current logon user ID will be used to generate the descriptor.

Request content

Your request must include a JSON object that describes the security resources to be validated. For additional details, see [Table 332 on page 456](#).

Table 332. Request content used to provision security resources defined in an HTTP request body.			
Field name	Type	Required or optional	Description
resourceItems	Object[]	Required	Array of security resources to be defined and permitted for the specified user ID.
resourceProfile	String	Required	Name of the security resource to be defined and permitted.
resourceClass	String	Required	Name of the class for the security resource to be defined and permitted.
access	String	Required	Access level of the security resource to be permitted for the specified user ID. Value can be set with the following: <ul style="list-style-type: none">• READ• UPDATE• CONTROL• ALTER

Authorization requirements

The user must be a security administrator.

Response content

On successful completion, HTTP status code 200 (OK) is returned. The response also includes a JSON object that contains the requested information. For additional details, see [Table 333 on page 456](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message. For additional details, see [“Error handling” on page 447](#).

Table 333. Response content to provision security resources defined in an HTTP request body.			
Field name	Type	Required or optional	Description
resourceItems	Object[]	Required	Array of the security resources validated for the specified user ID.
resourceProfile	String	Required	Name of the security resource profile. At the current stage: <ul style="list-style-type: none">• Variable in the name is not supported.

Table 333. Response content to provision security resources defined in an HTTP request body. (continued)

Field name	Type	Required or optional	Description
resourceClass	String	Required	SAF resource class.
access	String	Required	Level of access that is required for the security resource of the specified user ID or group ID. Value can be the following: <ul style="list-style-type: none"> • READ • UPDATE • CONTROL • ALTER
action	String	Required	For the provision action, the return value will be 'validate' or 'provision'. 'validate' means that the validation to the resource has a status of passed and no provision is conducted to it. 'provision' means that the validation to the resource has a status of failed or unknown and a provision is conducted to it.
actionObjectId	String	Required	The object ID of this action. For <i>provision</i> action, this ID is the user ID or group ID that is provisioned to the resource.
status	String	Required	Result of validation after the provision process. Value can be the following: <ul style="list-style-type: none"> • passed • failed • unknown
additionalInfo	String	Optional	Indicates additional information.
messageId	String	Optional	Indicates the message ID.
messageText	String	Optional	Indicates the message text.
httpStatus	Integer	Optional	Indicates the HTTP status code if an error is encountered.
requestMethod	String	Optional	Indicates the HTTP request method if an error is encountered.
requestUri	String	Optional	Indicates the URI if an error is encountered.

Example

In the following example, the POST method is used to provision security resources.

```

POST /zosmf/config/security/v1/provision?authid=ibmtest
{
  "resourceItems": [{
    "resourceProfile": "IZUDFLT.ZOSMF2",
    "resourceClass": "ZMFAPLA",
    "access": "READ"
  }, {
    "resourceProfile": "IZUDFLT.ZOSMF3",
    "resourceClass": "ZMFAPLA",
    "access": "READ"
  }]
}

```

Figure 199. Sample request body to validate security resources.

Figure 200 on page 458 provides a sample response from the validate security resources request.

```

{
  "resourceItems": [{
    "resourceProfile": "IZUDFLT.ZOSMF2",
    "resourceClass": "ZMFAPLA",
    "access": "READ",
    "action": "validate",
    "actionObjectId": "ibmtest",
    "status": "passed",
  }, {
    "resourceProfile": "IZUDFLT.ZOSMF3",
    "resourceClass": "ZMFAPLA",
    "access": "READ",
    "action": "provision",
    "actionObjectId": "ibmtest",
    "status": "failed",
    "additionalInfo": "Failed because ..."
  }]
}

```

Figure 200. Sample response body from the provision security resources request.

Software management services

The software management REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. This interface allows a client application to interact with the z/OSMF Software Management task.

Table 334 on page 458 lists the operations that the software management services provide.

Table 334. Operations provided through the software management services.

Operation	HTTP method and URI path
<u>“List the software instances defined to z/OSMF” on page 461</u>	GET /zosmf/swmgmt/swi
<u>“Retrieve the properties of a software instance” on page 464</u>	GET /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
<u>“List the data sets included in a software instance” on page 470</u>	POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/datasets

Table 334. Operations provided through the software management services. (continued)

Operation	HTTP method and URI path
<u>“Add a new software instance” on page 476</u>	POST /zosmf/swmgmt/swi
<u>“Export a defined software instance” on page 482</u>	POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/export
<u>“Modify the properties of a software instance” on page 488</u>	PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
<u>“Load the products, features, and FMIDs for a software instance” on page 495</u>	PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/products
<u>“Delete a software instance” on page 501</u>	DELETE /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
<u>“Deleting the Temporary Catalog Aliases” on page 502</u>	POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/deltempcatalias
<u>“List the portable software instances defined to z/OSMF” on page 507</u>	GET /zosmf/swmgmt/pswi
<u>“Deploy a software instance” on page 509</u>	POST /zosmf/swmgmt/dep
<u>“Delete a Deployment” on page 519</u>	DELETE /zosmf/swmgmt/dep/<dep-name>
<u>“Retrieve the properties of a portable software instance” on page 520</u>	GET /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>

Table 334. Operations provided through the software management services. (continued)

Operation	HTTP method and URI path
“Add a new portable software instance” on page 525	POST /zosmf/swmgmt/pswi
“Delete a portable software instance” on page 530	DELETE /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
“Retrieve the z/OS system UUID” on page 544	POST /zosmf/swmgmt/system/uuid/<system-nickname>

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. Some errors might also include a returned JSON object that contains the following attributes:

```
{
  "error":
  {
    "reason": "reason-code",
    "messages": ["message-text"]
  }
}
```

where:

error

JSON object that contains a reason code and a list of one or more message strings to describe the errors detected while processing the request.

reason-code

Reason code returned for the request. The value is an integer.

message-text

Array that contains the text of each message that was issued.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 401 Unauthorized

The submitter of the request did not authenticate to z/OSMF.

HTTP 403 Forbidden

The server rejected the request.

HTTP 404 Not found

The target of the request was not found.

HTTP 409 Conflict

The request could not be completed because there is a conflict with the current state of the resource.

HTTP 500 Internal server error

The server encountered an error that prevented it from completing the request.

HTTP 503 Service unavailable

The server is currently unavailable to process the request.

Error logging

Errors from the software management services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

List the software instances defined to z/OSMF

You can use this operation to obtain a list of the software instances that are defined to a z/OSMF instance.

HTTP method and URI path

```
GET /zosmf/swmgmt/swi
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, to obtain a list of the software instances that are defined to a z/OSMF instance, the user ID initiating the request must have READ access to the z/OSMF Software Management task. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 460.

If the request is successful, the response also includes the following JSON object:

```
{
  "swilist": [
    {
      "name": "swi-name",
      "system": "system-nickname",
      "description": "swi-description",
      "globalzone": "global-zone",
      "targetzones": ["target-zones"],
      "categories": ["categories"],
      "productinfo": "last-retrieved",
      "lastmodified": "last-modified",
      "modifiedby": "modified-user-ID",
      "created": "date-created",
      "createdby": "created-user-ID",
      "locked": "date-locked",
      "lockedby": "locked-user-ID",
      "swiurl": "swi-URL",
      "uuid": "swi-uuid"
    }
  ]
}
```

where:

swilist

Array that contains each software instance that is defined to z/OSMF.

swi-name

Name of the software instance.

system-nickname

Nickname of the z/OSMF host system that has access to the volumes and data sets where the software instance resides. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services”](#) on page 687.

swi-description

Description of the software instance.

global-zone

CSI data set that contains the global zone used to manage the software. If the software instance has no global zone, then this property will be null.

target-zones

Comma-separated list of the target zones included in the software instance. If the software instance has no global zone, then this property will be null.

categories

Comma-separated list of the categories to which the software instance is assigned.

last-retrieved

Date and time the product, feature, and FMID information was last retrieved for the software instance. This attribute is blank if this information has not been retrieved.

last-modified

Date and time in ISO 8601 format that the software instance was last modified.

modified-user-ID

User ID of the user who last modified the software instance.

date-created

Date and time in ISO 8601 format that the software instance was created.

created-user-ID

User ID of the user who created the software instance.

date-locked

Date and time in ISO 8601 format that the software instance was locked. This attribute is null if the software instance is not currently locked.

locked-user-ID

User ID of the user who locked the software instance. This attribute is null if the software instance is not currently locked.

swi-URL

URL that allows you to access the software instance. For example, a client application can use the URL to read a software instance. For more details, see [“Retrieve the properties of a software instance” on page 464](#).

swi-uuid

The UUID that is assigned to this software instance. A UUID is assigned to every software instance.

Example

In the following example, the GET method is used to retrieve a list of the software instances that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/swmgt/swi HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 201. Sample request to retrieve a list of software instances

A sample response is shown in [Figure 202 on page 463](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close

{"swiulist": [
  {"name": "DB2V9", "system": "PEV174", "description": null,
    "globalzone": "DB2.GLOBAL.CSI", "targetzones": ["DB2TGT"], "categories": null,
    "productinfo": "2014-08-20T19:23:25+00:00", "lastmodified": "2014-08-20T19:23:25+00:00",
    "modifiedby": "FRED", "created": "2014-08-20T19:23:25+00:00",
    "createdby": "BARNEY", "locked": null, "lockedby": null,
    "swiurl": "https://zosmf1.yourco.com/zosmf/swmgt/swi/bbc9e8d6-bd61-4f11-af48-ff500fffc178",
    "uuid": "bbc9e8d6-bd61-4f11-af48-ff500fffc178"},
  {"name": "zOSV2R1", "system": "PEV174", "description": null,
    "globalzone": "ZOS.GLOBAL.CSI", "targetzones": ["MVST100", "MVST110"],
    "categories": null, "productinfo": "2014-08-20T19:23:25+00:00",
    "lastmodified": "2014-08-20T19:23:25+00:00", "modifiedby": "WILMA",
    "created": "2014-08-20T19:23:25+00:00", "createdby": "BETTY", "locked": null,
    "lockedby": null, "swiurl": "https://zosmf1.yourco.com/zosmf/swmgt/swi/bbc9e8d6-bd61-4f11-af48-ff500fffc179",
    "uuid": "bbc9e8d6-bd61-4f11-af48-ff500fffc179"}
]}
```

Figure 202. Sample response from a request to retrieve a list of software instances

Retrieve the properties of a software instance

You can use this operation to retrieve the properties of a software instance. The properties include, but are not limited to, the global zone and target zones that are associated with the software instance and a list of the products, features, FMIDs, and non-SMP/E managed data sets that are included in the software instance.

HTTP method and URI path

```
GET /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

```
GET /zosmf/swmgmt/swi/<swi-uuid>
```

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify the software instance.

- **swi-uuid** further qualifies the request and indicates the specific software instance to be retrieved. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained using the [“List the software instances defined to z/OSMF” on page 461](#) REST API.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when you perform an analogous operation that uses the z/OSMF Software Management task. That is, to retrieve the properties of a software instance, the user ID initiating the request must have READ access to both the z/OSMF Software Management task and the software instance. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 460](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "uuid": "swi-uuid",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "productinfo": {
    "retrieved": "last-retrieved",
    "lastmodified": "last-modified",
    "modifiedby": "modified-user-ID",
    "created": "date-created",
    "createdby": "created-user-ID",
    "locked": "date-locked",
    "lockedby": "locked-user-ID",
    "datasets": [
      {
        "dsname": "data-set-name",
        "volume": "volume-serial"
      }
    ]
  },
  "products": [
    {
      "prodname": "product-name",
      "prodid": "product-ID",
      "release": "product-level",
      "vendor": "vendor-name",
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service",
      "url": "product-URL",
      "productinfo": {
        "fileversion": "file-version",
        "features": [
          {
            "feature": "feature-name",
            "fmids": [
              {
                "fmid": "fmid-name",
                "description": "fmid-description",
                "targetzones": ["fmid-target-zones"]
              }
            ]
          }
        ]
      }
    }
  ],
  "nonsmpeproducts": [
    {
      "prodname": "product-name",
      "prodid": "product-id",
      "release": "product-level",
      "vendor": "vendor-name",
      "url": "product-url",
      "features": ["feature-name"],
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date"
    }
  ],
  "workflows": [
    {
      "name": "workflow-name",
      "description": "workflow-description",
      "location": {
        "smptype": "smp-type",
        "smpname": "smp-name",
        "dsname": "workflow-dsname",
        "path": "workflow-path"
      },
      "performonhosts": true | false
    }
  ]
}
```

```

    },
    "datasetproperties": [{
      "dddefname": "dddef-name",
      "zone": "zone-name",
      "dsname": "data-set-name",
      "volume": "volume-serial",
      "dstype": "DLIB",
      "properties": [{"key": "value"}]
    }],
    "datasetpropertylabels": [{
      "propertyname": "property-name",
      "label": "property-label"
    }],
    "productproperties": [{
      "prodid": "product-id",
      "release": "product-level",
      "prodname": "product-name",
      "properties": [{"key": "value"}]
    }]
  }

```

Where:

swi-name

Name of the software instance.

swi-uuid

The UUID that is assigned to this software instance. A UUID is assigned to every software instance.

system-nickname

Nickname of the z/OSMF host system that has access to the volumes and data sets where the software instance resides. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

swi-description

Description of the software instance.

global-zone

CSI data set that contains the global zone used to manage the software. If the software instance has no global zone, then this property will be null.

target-zones

Comma-separated list of the target zones included in the software instance. If the software instance has no global zone, then this property will be null.

categories

Comma-separated list of the categories to which the software instance is assigned.

last-retrieved

Date and time the product, feature, and FMID information was last retrieved for the software instance. This attribute is blank if this information has not been retrieved.

last-modified

Date and time in ISO 8601 format that the software instance was last modified.

modified-user-ID

User ID of the user who last modified the software instance.

date-created

Date and time in ISO 8601 format that the software instance was created.

created-user-ID

User ID of the user who created the software instance.

date-locked

Date and time in ISO 8601 format that the software instance was locked. This attribute is null if the software instance is not currently locked.

locked-user-ID

User ID of the user who locked the software instance. This attribute is null if the software instance is not currently locked.

datasets

Array that contains each non-SMP/E managed data set that is included in the software instance.

data-set-name

Name of the non-SMP/E managed data set.

volume-serial

Volume on which the non-SMP/E managed data set resides.

products

Array that contains each product that is included in the software instance.

product-name

Name of the product. If any FMIDs are not related to a product and feature, those FMIDs are listed under a product named *No Product*.

product-ID

Identifier of the product.

product-level

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

vendor-name

Name of the vendor that provides the product.

general-availability

Date this level of the product is available to all users.

end-of-service

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions.

product-URL

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions.

file-version

Version of the most recent product information file that was retrieved that contains the corresponding product. The version represents the date that file was created or last updated.

features

Array that contains each feature that is included in the product.

feature-name

Name of the feature. If any FMIDs are not related to a product and feature, those FMIDs are listed under a feature named *No Feature*.

fmids

Array that contains each FMID that is included in the feature.

fmid-name

Name of the FMID.

fmid-description

Description of the FMID.

fmid-target-zones

Name of the target zones where the FMID is installed.

nonsmpeproducts

List of products for the software instance that are not managed by SMP/E.

product-name

Name of the product, but can be up to 64 characters.

product-ID

Identifier for the product, but can be up to 64 characters.

product-level

Release level for the product, but can be up to 64 characters.

vendor-name

Name of the vendor that provides the product, but can be up to 64 characters.

product-URL

A URL that links to additional information about the product, but can be up to 256 characters.

feature-name

List of names of features for the product, but can be up to 64 characters.

general-availability-date

Date this level of the product is available to all users.

end-of-service-date

Last date on which vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

null

The end of service date is unknown for the product.

yyyy-mm-ddThh:mm:ssZ

The known end of service date, in ISO 8601 format.

NotAnnounced

The end of service date is not yet announced for the product.

workflows

List of workflows for the software instance.

workflow-name

Name for the workflow.

workflow-description

Description for the workflow.

location

Location of the workflow definition file for the workflow.

smp-type

The SMP/E element type for a workflow definition file that is managed by SMP/E.

smp-name

The SMP/E element name for a workflow definition file that is managed by SMP/E.

workflow-dsname

The name of the data set that contains the workflow definition file.

workflow-path

The UNIX path for a workflow definition file that is a UNIX file.

performonhostsystm

Indicates whether the workflow steps can be performed on the host system or on another system.

true

Indicates that the workflow steps can be performed on the z/OSMF host system on which the software instance resides.

false

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

datasetproperties

A list of one or more properties for individual data sets.

dddefname

The name of the SMP/E DDDEF entry that describes an SMP/E managed data set.

zone

The zone name where the DDDEF entry resides.

dsname

The name of the subject data set.

volume

The volume of the subject .

dstype

The usage type of the subject data set. A value of DLIB indicates that the data set is the dataset is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone.

properties

A list of one or more properties for the subject data set, which is specified, as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean array, object or null.

datasetpropertylabels

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

propertyname

The name, or key, of the existing provider defined property.

label

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

productproperties

A list of one or more properties for individual software products.

prodid

The identifier for the subject product.

release

The version, release, modification level for the subject product.

prodname

The name of the subject product.

properties

A list of one or more properties for the subject product, which is specified as key-value pairs. The keys are strings, and the values are a valid JSON data type such as string, number, Boolean, array, object or null. See [“Provider Defined Properties” on page 1302](#) for more information.

Example

In the following example, the GET method is used to retrieve the properties of software instance DB2V9 on system PEV174.

```
GET /zosmf/swmgt/swi/bbc9e8d6-bd61-4f11-af48-ff500fffc178 HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 203. Sample request to retrieve the properties of a software instance

A sample response is shown in [Figure 204 on page 470](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close

{
  "name": "DB2V9", "system": "PEV174", "description": null, "uuid": "bbc9e8d6-bd61-4f11-af48-ff500fffc178",
  "globalzone": "DB2.GLOBAL.CSI", "targetzones": ["DB2TGT"], "categories": null,
  "productinfo": { "retrieved": "2014-08-20T19:23:25+00:00", "lastmodified": "2014-08-20T19:23:25+00:00",
    "createdby": "BARNEY", "locked": null, "lockedby": null, "datasets": [
      { "dsname": "USER.DB2V9.PROCLIB", "volume": "LV1234" },
      { "dsname": "USER.DB2V9.SAMPLES", "volume": "LV1234" } ], "products": [
      { "prodname": "DB2 for z/OS", "prodid": "5635-DB2", "release": "09.01.00",
        "vendor": "IBM", "generalavailability": "20006-06-09T19:23:25+00:00",
        "endofservice": "2014-06-27T19:23:25+00:00", "url": null,
        "productinfo": { "fileversion": "2014-01-01", "features": [ { "feature": "DB2 Base",
          "fmids": [ { "fmid": "HDB9910", "description": "DB2 BASE/TSO", "targetzones": [
            "DB2V9T" ] } ] } ] } ] } ] }
}

```

Figure 204. Sample response from a request to retrieve the properties of a software instance

List the data sets included in a software instance

You can use this operation to obtain a list of the data sets that compose a software instance.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/datasets
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/datasets
```

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of **<system-nickname/swi-name>** to identify the software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the [“List the software instances defined to z/OSMF” on page 461](#) REST API.
- **datasets** indicate that the data sets included in the software instance are to be obtained.

When you issue this request, z/OSMF analyzes the global, target, and distribution zones included in the software instance to identify the SMP/E managed data sets and the SMP/E managed UNIX data sets that contain the installed software that is described in those zones. z/OSMF returns a JSON object that lists the properties of each data set it identified, along with the properties of each non-SMP/E managed data set included in the software instance.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content that is provided by the caller. Use the JSON content type ("Content-Type: application/json") if a JSON document is included as input with this request.

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

If the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. To do so, include the following JSON object in your request:

```
{
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 205. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server

Where:

zosmf-user-ID

User ID for authenticating with the secondary z/OSMF instance.

zosmf-password

Password for authenticating with the secondary z/OSMF instance.

proxy-user-ID

User ID for authenticating with the HTTP proxy server.

proxy-password

Password for authenticating with the HTTP proxy server.

Include the JSON object in the request only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when you perform an analogous operation by using the z/OSMF Software Management task. That is, the user ID must have READ access to both the Software Management task and the SAF resources that are required to process the request. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request was accepted. If the request was accepted, the service returns status code 202 Accepted and a JSON object that contains a URL (`{"statusurl": "url"}`). To monitor the status of the list data sets request and to retrieve the results, perform GET requests to the supplied URL. Only the user ID that initiates the list data sets request is authorized to check the status and retrieve the results. One of the following responses is returned from the get status request:

- If the list data sets request is still in progress, an HTTP response code of 200 OK is returned, along with the following JSON object: `{"status": "status", "percentcomplete": "percent-complete"}`.
- If the list data sets request is complete, an HTTP response code of 200 OK is returned, along with the following JSON object:

```
{
  "status": "status",
  "percentcomplete": "percent-complete",
  "swidatasets": {
    "smpmanageddatasets": [{
      "dsname": "data-set-name",
      "volumes": ["volume-serial"],
      "dstype": "data-set-type",
      "recfm": "record-format",
      "lrecl": "logical-record-length",
      "blksize": "block-size",
      "tracks": "allocated-tracks",
      "used": "used-tracks-percent",
      "extents": "allocated-extents",
      "dscategory": ["data-set-category"],
      "zones": ["zones"],
      "zoneddefs": [{
        "zone": "zone-name", "dddefs": ["dddef-name"]}],
      "msgs": ["message-text"]
    }],
    "smpmanagedunixdatasets": [{
      "dsname": "data-set-name",
      "unixdirs": ["UNIX-directory"],
      "volumes": ["volume-serial"],
      "dstype": "data-set-type",
      "recfm": "record-format",
      "lrecl": "logical-record-length",
      "blksize": "block-size",
      "tracks": "allocated-tracks",
      "used": "used-tracks-percent",
      "extents": "allocated-extents",
      "dscategory": ["data-set-category"],
      "zones": ["zones"],
      "zoneddefs": [{
        "zone": "zone-name", "dddefs": ["dddef-name"]}],
      "msgs": ["message-text"]
    }],
    "nonsmpmanageddatasets": [{
      "dsname": "data-set-name",
      "volumes": ["volume-serial"],
      "dstype": "data-set-type",
      "recfm": "record-format",
      "lrecl": "logical-record-length",
      "blksize": "block-size",
      "tracks": "allocated-tracks",
      "used": "used-tracks-percent",
      "extents": "allocated-extents",
      "dscategory": ["data-set-category"],
      "unixdirs": ["UNIX-directory"],
      "msgs": ["message-text"]
    }],
  }
}
```

Where:

status

Status of the list data sets request. The status is either *running* or *complete*.

percent-complete

Percentage of the processing that is complete for the list data sets request, expressed as a whole number from 0 to 100.

swidatasets

Lists all the data sets included in the software instance. A software instance can contain:

- **smpmanageddatasets**. Array of the SMP/E managed data sets included in the software instance.
- **smpmanagedunixdatasets**. Array of the SMP/E managed UNIX file system data sets included in the software instance.
- **nonsmpmanageddatasets**. Array of the non-SMP/E managed data sets included in the software instance.

data-set-name

Name of the data set.

UNIX-directory

Array of the UNIX directories contained in the data set. If z/OSMF could not identify the UNIX file system data set for any UNIX directories, the data set name for those directories is set to *No Data Set Found*. Typically, this occurs when the UNIX file system data set is not mounted.

volume-serial

Array of the volume serials for the volumes on which the data set resides. If the data set is migrated, a value of *MIGRAT* is returned.

data-set-type

Type of data set. The data set can be one of the following types:

- **HFS**. Hierarchical file system.
- **PDS**. Partitioned data set.
- **PDSE**. Partitioned data set extended.
- **Sequential**. Sequential data set.
- **VSAM**. VSAM data set.
- **ZFS**. zSeries file system.

record-format

Record format specified when the data set was allocated. The record format can be any valid combination of the following codes:

- **A**. ASA printer control characters.
- **B**. Blocked records.
- **F**. Fixed-length records.
- **M**. Machine code printer control characters.
- **S**. Standard (for F) or spanned (for V); used only with sequential data sets.
- **T**. Track-overflow feature.
- **U**. Undefined format records.
- **V**. Variable-length records.

logical-record-length

Logical record length, in bytes, specified when the data set was allocated.

block-size

Block size, in bytes, specified when the data set was allocated.

allocated-tracks

Number of tracks allocated to the data set.

used-tracks-percent

Percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

allocated-extents

Number of extents that are allocated to the data set.

dscategory

List of categories for how the data set is used. Can be one or more of the following:

- TARGET - SMP/E managed target library, or SMP/E control data set associated with a target zone.
- DLIB - SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL - SMP/E control data set associated with the global zone.
- SMP/E - SMP/E control data set.
- WORKFLOW - Contains one or more workflow definition files for the workflows that are explicitly defined to the software instance.

zones

Array of the SMP/E zones that contain a DDDEF entry for the data set. For an SMPCSI data set, it is an array of the SMP/E zones that are contained in the data set.

zonedefs

Array of SMP/E zones and DDDEF entries for the data set.

zone-name

Name of an SMP/E zone that contains one or more DDDEF entries for the data set.

dddef-name

Array of DDDEF entry names that identify the data set.

message-text

Array of messages returned for the data set.

- If the list data sets request is complete but the results are no longer available, an HTTP response code of 404 Not found is returned. z/OSMF makes the results available for a client application for a finite period of time. When that time elapses, the results are no longer available; in which case, the client must reissue the request.

If the list data sets request cannot be processed, a status code of 4nn or 5nn is returned, indicating that an error has occurred. For more details, see [“Error handling” on page 460](#).

Example

In the following example, the POST method is used to retrieve a list of the data sets included in software instance *DB2V9* on system *SYS123*.

```
POST /zosmf/swmgmt/swi/SYS123/DB2V9/datasets HTTP/1.1
Host: sys123.yourco.com
```

Figure 206. Sample request to list the data sets included in a software instance

Figure 207 on [page 475](#) provides a sample response, which indicates that the list data sets request has been accepted and is supplying the URL to use for monitoring the status of that request.


```

HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgt/statusmonitor/dslist
/4837290198343"}

```

Figure 207. Sample response for a list data sets request

To check the status of the list data sets request, submit the following request:

```

GET /zosmf/swmgt/statusmonitor/dslist/4837290198343 HTTP/1.1
Host: sys123.yourco.com

```

Figure 208. Sample request to obtain the status of a list data sets request

Figure 209 on page 475 provides a sample response for when the list data sets request is in progress.

```

HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running", "percentcomplete":"65"}

```

Figure 209. Sample get status response when the list data sets request is in progress

Figure 210 on page 475 provides a sample response for when the list data sets request is complete.

```

HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "percentcomplete":"100", "swidatasets":{
  "smpmanageddatasets":[
    {"dsname":"USERID.SMPE.CSI", "volumes":["SL730C"], "dstype":"VSAM",
      "recfm":null, "lrecl":null, "blksize":null, "tracks":"1509", "used":null,
      "extents":null, "zones":["GLOBAL", "TGT0", "DLB0"], "zonedddefs":null,
      "msgs":null},
    {"dsname":"USERID.SMPE.MACLIB", "volumes":["SL7334"], "dstype":"PDS",
      "recfm":"FB", "lrecl":"80", "blksize":"27920", "tracks":"4", "used":"100",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["MACLIB", "SMPMTS"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.MIGLIB", "volumes":["SL8B2D"], "dstype":"PDS",
      "recfm":"U", "lrecl":null, "blksize":"32760", "tracks":"147", "used":"100",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["MIGLIB"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.SGIMCLS0", "volumes":["SL7307"], "dstype":"PDS",
      "recfm":"VB", "lrecl":"255", "blksize":"32760", "tracks":"15", "used":"6",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["SGIMCLS0"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.SGIMMJPN", "volumes":["MIGRAT"], "dstype":null,
      "recfm":null, "lrecl":null, "blksize":null, "tracks":null, "used":null,
      "extents":null, "zones":null, "zonedddefs":null, "msgs": ["GIM70531E Data set
      USERID.SMPE.SGIMMJPN is migrated and is being recalled."]}],
    ...],
  "smpmanagedunixdatasets":[{"dsname":"USERID.SMPE.ZFS", "unixdirs":
    ["/u/userid/smpe/usr/lpp/smp/IBM/"], "volumes":["ZF3804"], "dstype":"ZFS",
    "recfm":null, "lrecl":null, "blksize":null, "tracks":"22650", "used":null,
    "extents":null, "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
      ["SGIMBIN"]}]}], "msgs":null},
  "nonsmpmanageddatasets":null
}}

```

Figure 210. Sample get status response when the list data sets request is complete

Add a new software instance

You can use this operation to add a software instance to z/OSMF.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document included as input with this request.

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

Your request must include a JSON object that describes the software instance to be added, for example:

```

{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "datasets": [
    {
      "dsname": "data-set-name",
      "volume": "volume-serial"
    }
  ],
  "products": [
    {
      "prodname": "product-name",
      "prodid": "product-id",
      "release": "product-level",
      "vendor": "vendor-name",
      "url": "product-url",
      "features": ["feature-name"],
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date"
    }
  ],
  "workflows": [
    {
      "name": "workflow-name",
      "description": "workflow-description",
      "location": {
        "smptype": "smp-type",
        "smpname": "smp-name",
        "dsname": "workflow-dsname",
        "path": "workflow-path"
      },
      "performonhosts": true | false
    }
  ],
  "datasetproperties": [
    {
      "dddefname": "dddef-name",
      "zone": "zone-name",
      "dsname": "data-set-name",
      "volume": "volume-serial",
      "dstype": "DLIB",
      "properties": [{"key": "value"}]
    }
  ],
  "datasetpropertylabels": [
    {
      "propertyname": "property-name",
      "label": "property-label"
    }
  ],
  "productproperties": [
    {
      "prodid": "product-id",
      "release": "product-level",
      "prodname": "product-name",
      "properties": [{"key": "value"}]
    }
  ]
}

```

Figure 211. Adding a software instance: request content

where:

swi-name

Name of the software instance. The name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ _ # @ ^). The name is required and must be unique on the system.

system-nickname

Nickname of the system that has access to the volumes and data sets where the software instance resides. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems defined to z/OSMF, you can use the z/OSMF topology services. For more details, see [“Topology services”](#) on page 687.

swi-description

Description of the software instance. The description is optional and can contain a maximum of 256 characters.

global-zone

CSI data set that contains the global zone used to manage the software. If specified, must comply with z/OS data set naming conventions, and must end with .CSI.

target-zones

Comma-separated list of the target zones to be included in the software instance. A list of 0 or more zone names, each target zone name must be 1-7 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and special characters (@ # \$). The first character must be an alphabetic character.

categories

Comma-separated list of the categories to which the software instance is assigned. Each category name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ _ # @ ^). Assigning the software instance to a category is optional.

datasets

Array that contains each non-SMP/E managed data set to be added to the software instance. Adding non-SMP/E managed data sets to the software instance is optional.

data-set-name

Name of the non-SMP/E managed data set. You cannot specify data set members or a subset of a data set. A data set name is required if you are adding non-SMP/E managed data sets to the software instance. The data set name must comply with z/OS data set naming conventions

volume-serial

Volume on which the non-SMP/E managed data set resides. The volume serial is required if the dataset is not cataloged, and the volume must be accessible by the system where the software instance resides. The volume serial must be 1-6 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

products

List of products for the software instance that are not managed by SMP/E. This list is optional.

product-name

Name of the product, but can be up to 64 characters.

product-ID

Identifier for the product, but can be up to 64 characters.

product-level

Release level for the product, but can be up to 64 characters.

vendor-name

Name of the vendor that provides the product, but can be up to 64 characters.

product-URL

A URL that links to additional information about the product, but can be up to 1023 characters.

feature-name

List of names of features for the product, but can be up to 64 characters.

general-availability-date

Date this level of the product is available to all users. May be null, or a date value, in ISO 8601 format, yyyy-mm-ddThh:mm:ssZ.

end-of-service-date

Last date on which the vendor will deliver standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

null

The end of service date is unknown for the product.

yyyy-mm-ddThh:mm:ssZ

The known end of service date, in ISO 8601 format.

NotAnnounced

The end of service date is not yet announced for the product.

workflows

List of workflows for the software instance. This list is optional.

workflow-name

Name for the workflow. The name may contain up to 100 characters, but must not include the characters for ampersand ('&'), forward slash ('/'), backward slash ('\'), logical or ('|'), greater than ('>'), or less than ('<'). Embedded blanks are allowed.

workflow-description

Description for the workflow. The description is optional, and can contain a maximum of 256 characters.

location

Location of the workflow definition file for the workflow. A workflow definition file location is required.

smp-type

The SMP/E element type for a workflow definition file that is managed by SMP/E. The SMP/E element type is optional and may be up to 12 uppercase alphanumeric characters. The first character cannot be numeric.

smp-name

The SMP/E element name for a workflow definition file that is managed by SMP/E. The SMP/E element name is optional and may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

workflow-dsname

The sequential data set name, or partitioned data set name and member for a workflow definition file that is in a data set. The data set name is optional, but if specified must comply with z/OS data set naming conventions. For example, IBMUSR.SM.WRKFLOWS(WRKFLOW1).

workflow-path

The UNIX path for a workflow definition file that is a UNIX file. The UNIX path is optional, but if specified it must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash.
- Can be up to 1023 characters long.

performonhostsystm

Indicates whether the workflow steps may be performed on the host system or on another system. This property is optional and the default value is true.

true

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

false

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

Tip: Ensure that the description property contains information that helps the user select an appropriate alternative system for the performance of the workflow steps. For example, if the workflow steps are for installation verification procedures (IVPs), the user would select the system on which the newly activated software runs.

datasetproperties

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,” on page 1297](#) for more information.

To define properties for SMP/E managed data sets, specify the DDDEF entry name that identifies the desired data set or data sets. Specify the zone name only when necessary to identify a unique data set. For example, if there are more than one DDDEF entries with the same name pointing to different data sets, then both a DDDEF entry name and zone name are required to define properties for only one of those data sets. If an SMP/E zone name is not specified then the properties apply to all data sets identified by all DDDEF entries with the same name in all zones that reside in the software instance.

To define properties for a non-SMP/E managed data set, specify the data set name and the volume for the data set as they are used when defining the data set to the software instance. That is, if only a data set name is specified to define the data set to the software instance, then specify only the data set name when defining properties for the data set. If a data set name and volume are specified to define the data set to the software instance, then specify both the data set name and volume to define properties for the data set.

dddefname

The name of the SMP/E DDDEF entry that describes one or more SMP/E managed data sets. It may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

zone

The zone name where the DDDEF entry resides. It may be up to 7 uppercase alphanumeric, \$, #, or @ characters.

dsname

The name of the subject data set. This is used to identify a non-SMP/E managed data set. It may be up to 44 characters and must comply with z/OS data set naming conventions.

volume

The volume of the subject data set. This is used to identify a non-SMP/E managed data set where the volume had been specified to identify an uncatalogued data set. It may be up to 6 uppercase alphanumeric, \$, #, or @ characters.

dstype

The usage type of the identified data set. The value may be DLIB or the default value of null. A value of DLIB indicates the identified data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone. The dstype value of DLIB must be specified for properties that identify distribution library or related control data sets so Software Management can safely ignore this property specification when a software instance does not contain SMP/E managed distribution libraries or related control data sets.

properties

A list of one or more properties for the subject data set, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

datasetpropertylabels

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

propertyname

The name, or key, of the existing provider defined property.

label

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

productproperties

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 1297 for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

prodid

The identifier for the subject product. This is required to identify an SMP/E managed product. The product identifier may be up to 64 characters.

release

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product. The release level may be null, or up to 64 characters.

prodname

The name of the subject product. This is required to identify a non-SMP/E managed product. The product name may be null, or up to 64 characters.

properties

A list of one or more properties for the subject product, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task, and CONTROL access to the SAF resources corresponding to the software instance to be added. If the specified categories are implicitly added during this software instance add operation, the user ID must also have CONTROL access to the SAF resources corresponding to the specified categories. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 460.

Example

In the following example, the POST method is used to add a software instance to the z/OSMF instance that has a host name of *pev174.yourco.com*.

```
POST /zosmf/swmgmt/swi HTTP/1.1
Host: pev174.yourco.com Content-Type: application/json Accept-Language:
en { "name": "Hooli", "system": "PEV174", "description": "Fictitious software
product from Hooli", "globalzone": "IBMUSER.HOOLI.CSI", "targetzones":
[ "TGTZ", "datasets": [ { "dsname": "IBMUSER.HOOLI.PROCLIB", "volume": "LV1234",
{ "dsname": "IBMUSER.HOOLI.PARMLIB", "volume": "LV1234" } ], "products": [ { "prodname": "Hooli
Express", "release": "2.1", "vendor": "Hooli Software" } ], "workflows": [ { "name": "Setup
Hooli",
"description": "Setup and configure fictitious software product
Hooli",
"location": { "dsname": "IBMUSER.HOOLI.WORKFLOW(SETUP)" } }, { "datasetproperties":
[ { "dddefname": "HOOLMOD", "properties": [ { "hooli-lnkLst": "yes", { "hooli-ispfType": "LMOD" } } ],
{ "dddefname": "HOOLMSG", "properties": [ { "hooli-ispfType": "MESSAGE" } } ],
{ "dddefname": "HOOLPNL", "properties": [ { "hooli-ispfType": "PANEL" } } ],
{ "dddefname": "HOOLSKL", "properties": [ { "hooli-ispfType": "SKELETON" } } ],
{ "dddefname": "HOOLTBL", "properties": [ { "hooli-ispfType": "TABLE" } } ],
{ "dddefname": "HOOLREXX", "properties": [ { "hooli-ispfType": "EXEC" } } ], "datasetpropertylabels":
[ { "propertyname": "hooli-lnkLst", "label": "LNKLST Eligible", { "propertyname": "hooli-
ispfType", "label": "ISPF Element Type" } ] }
```

Figure 212. Sample request to add a software instance

Export a defined software instance

A portable software instance is a set of portable files that represents the content of a z/OSMF software instance. An Export action on a software instance is used to create a portable software instance. You can use the POST method to perform an Export action on a software instance that is defined to z/OSMF, which generates a portable software instance descriptor file and JCL that when executed creates the archive files for a portable software instance, and store those files in a UNIX directory on the system where the software instance being exported resides.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/export
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/export
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be exported. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be exported. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).
- **/export** indicates JCL to perform an export action for the software instance is to be generated.

Standard headers

Use the following standard HTTP headers with this request:

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type("Content-Type: application/json") is used for the JSON document, if any, included as input with this request.

Custom headers

None.

Request content

The request must include a JSON document that identifies properties that are required to perform the operation. For example,:


```
{
  "packagedir": "UNIX-directory",
  "jcldataset": "data-set-name",
  "includedlibs": "yes | no",
  "datasetmergeallowed": "yes | no",
  "signingcertificate": "certificate-label",
  "keyring": "keyring-name" | "userid/keyring-name",
  "jobstatement": ["jclrecord"],
  "unixdatasets": [
    {
      "dsname": "data-set-name",
      "mountpoint": "UNIX-path"
    }
  ],
  "workdsnprefix": "data-set-name-prefix",
  "workvolume": "volume-serial",
  "workstorclas": "storage-class",
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

where:

packagedir

UNIX directory to contain the files for the portable software instance.

Must be a UNIX directory with valid UNIX directory name syntax:

- Must be an absolute pathname (must start with slash).
- Must end with a slash.
- Can be up to 1023 characters long.

jcldataset

Partitioned data set to contain the portable software instance descriptor file and the generated JCL.

The data set name must comply with z/OS data set naming conventions.

includedlibs

If the software instance contains SMP/E managed software, this property indicates whether the distribution libraries and DLIB zones are to be included in the portable software instance.

This is an optional property. Can be null, "yes" or "no", but the default value is "yes".

datasetmergeallowed

Indicates whether the merging of the data sets is allowed when you deploy the portable software instance. This is an optional property. Can be "null", "yes" or "no". The default value is "no".

signingcertificate

The label to identify the certificate that is used to sign the portable software instance. This certificate is bound to the private key that is used to sign the GIMZIP package. It contains the matching public key that is used to verify the signature later. This is an optional property, but if specified, the portable software instance is digitally signed.

Certificate labels are from 1 to 32 characters in length and may include only characters X'40' through X'FE', excluding the reserved XML characters, less than (<), greater than (>), double quotation mark ("), and ampersand (&).

If you specify `signingcertificate`, then you must also specify `keyring`.

keyring

The security manager key ring that contains the specified signing certificate. The keyring must also contain all certificates in the certification path from the signing certificate to the issuing root CA certificate.

Key ring names are from 1 to 237 characters in length and may include only characters X'40' through X'FE', excluding the forward slash (/) and the reserved XML characters, less than (<), greater than (>), double quotation mark ("), and ampersand (&).

If the key ring is associated with a user ID other than the one that is used to execute the program GIMZIP, then the key ring name must be prefixed with the associated user ID. User IDs are from 1 to 8 alphanumeric characters in length, and can consist entirely of numbers and need not begin with any particular character. The user ID is separated from the key ring value by a forward slash (ie. userid/keyring).

If you specify keyring, then you must also specify signingcertificate.

jobstatement

List of JCL cards for the JOB statement to be used in the generated JCL for the export operation.

This is an optional property. Can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 to 8 characters. If no JOB statement is provided, the default is exactly://IZUD01EX JOB (ACCOUNT),'NAME'.

unixdatasets

List of UNIX file system data sets in the software instance that are currently not mounted, and therefore, cannot be identified by z/OSMF by referencing the UNIX directories that are defined in the SMP/E target zone DDDEF entries.

This is an optional property, required only if the software instance describes SMP/E managed software, and if any of the UNIX file system data sets containing that software are currently not mounted.

dsname

Name of a UNIX file system data set.

The data set name must comply with z/OS data set naming conventions.

mountpoint

Mount point for the UNIX file system data set, if the data set were mounted.

Must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash, unless a slash is the only character. That is, "/" (root) is valid, but "/mountpoint/" is not.
- Can be up to 1023 characters long.

workdsnprefix

Data set name prefix for the work data sets that are created, used, and then deleted in the generated Export JCL.

z/OSMF uses a work zFS data set to contain temporary UNIX directory space. The prefix can contain up to 26 characters, including periods, and must comply with z/OS data set naming conventions. This value is optional.

Notes:

- If one or more of the workdsnprefix, workvolume, or workstorclas values are specified, then the generated JCL creates, mounts, uses, unmounts, and deletes a work zFS data set. If none of those three values are specified, then the generated JCL does not create and use a work zFS data set and instead uses /tmp as the temporary UNIX directory space.
- If you specify workdsnprefix but do not provide values for either workvolume or workstorclas, then the volume and storclas are not specified in the generated JCL on the IDCAMS DEFINE CLUSTER command to create the zFS data set.
- If workdsnprefix is not specified but either workvolume or workstorclas is provided, then a default data set name prefix of z/OSMF-ID.WRKDSN is used, where z/OSMF-ID is the logged-in user ID that is used to authenticate to the z/OSMF server.

workvolume

Volume for the work data sets that are created, used, and then deleted in the generated Export JCL.

The volume serial value must contain 1-6 characters. Valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

Notes:

- See [“workdsnprefix”](#) on page 484 for notes that also apply here.
- The workvolume and workstorclas properties are mutually exclusive.

workstorclas

Storage class for the work data sets that are created, used, and then deleted in the generated Export JCL.

The storage class value can contain up to eight characters. Valid characters are alphabetic characters (A-Z, a-z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

Notes:

- See [“workdsnprefix”](#) on page 484 for notes that also apply here.
- The workvolume and workstorclas properties are mutually exclusive.

zosmfuid

Userid for authenticating with a remote z/OSMF instance.

This is an optional property.

zosmfpw

Password for authenticating with a remote z/OSMF instance.

This is an optional property

proxyuid

Userid for authenticating with an HTTP proxy.

This is an optional property.

proxypw

Password for authenticating with an HTTP proxy.

This is an optional property.

The request content is required, but some properties are optional. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. Therefore, you can need to specify the remote z/OSMF userid, password, and proxy userid and password.

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation that uses the z/OSMF Software Management task. That is, to export a software instance, the user ID must have READ access to the Software Management task and READ access to the SAF resources for the software instance being exported. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

Generating JCL to export a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document containing a URL for the status monitor for the request.

The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example:

```
{
  "statusurl": "url"
}
```

where:

statusurl

Indicates the URL that provides status for the software instance export request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document containing status information for the operation.
- If the operation has completed, then an HTTP response code of 200 OK is returned, along with a JSON document containing status information and the desired result set.
- If the request has expired, then an HTTP response code of 404 Not found is returned. That is, when the operation has completed, the z/OSMF server holds the result set for a finite length of time. After that time has passed, the result set is said to expire and will no longer be available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",
  "percentcomplete": "percent",
  "jcl": ["data-set-name(member-name)"]
}
```

status

The status of the export request. The status can be either "running" or "complete".

percentcomplete

The percentage of the processing that is complete for this export software instance request, expressed as a whole number from 0 to 100.

jcl

Ordered list of generated jobs for the export action. Each job is a unique member in the JCL data set. The values are fully qualified data set and member names of the form "data-set-name(member-name)".

See “Error handling” on page 460 for the error response document containing a reason code and a list of one or more message strings to describe the errors that are detected during processing of a request.

Example

The following request generates an Export job for the software instance that is named *DB2V9* on *SYS123*.

```
POST /zosmf/swmgmt/swi/SYS123/DB2V9/export HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "packagedir": "/u/userid/exportDir/",
  "jcldataset": "USERID.SMJCL.CNTL",
  "includedlibs": "yes",
  "jobstatement": ["/EXPORT JOB (123456), 'USER', NOTIFY=&SYSUID,",
  "/ MSGCLAS=H", "/ /*"]
}
```

Figure 213. Sample request

A sample response is as follows:

```
HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgt/statusmonitor/export/4837290198343"}
```

Figure 214. Sample response

The above response indicates the request to generate an Export job for the software instance has been accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is as follows:

```
GET /zosmf/swmgt/statusmonitor/export/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response is as follows:

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running", "percentcomplete":"65"}
```

Figure 215. Sample response

The above response indicates the operation to generate an Export job for the software instance is still running and 65% complete. A final request to the status monitor URL is as follows:

```
GET /zosmf/swmgt/statusmonitor/export/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response is as follows:

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "percentcomplete":"100",
 "jcl":["USERID.SMJCL.CNTL(IZUD01EX)"]}
}
```

Figure 216. Sample response

Usage notes

The POST method to generate JCL that exports a software instance requires the subject software instance to be already defined to z/OSMF. If the software instance is not already defined, you can use the POST method to Add a new software instance to z/OSMF, followed by the PUT method to Load the products, features, and FMIDs for the new software instance if that software instance contains SMP/E managed software. Then, you can use the POST method to Export the software instance. For example:

Add the software instance:

```
POST /zosmf/swmgt/swi HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "name":"DB2V9",
  "system":"sys123",
  "description":"DB2",
}
```

```
"globalzone": "DB2.GLOBAL.CSI",
"targetzones": ["DB2TGT"],
}
```

Load the products, features, and FMIDs for the software instance:

```
PUT /zosmf/swmgmt/swi/sys123/DB2/products HTTP/1.1
Host: sys123.yourco.com
```

Create JCL to export the software instance:

```
POST /zosmf/swmgmt/swi/sys123/DB2/export HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "packagedir": "/u/userid/exportDir/",
  "jcldataset": "USERID.SMJCL.CNTL",
  "includedlibs": "yes",
  "jobstatement": ["//EXPORT JOB (123456), 'USER', NOTIFY=&SYSUID, ",
  "// MSGCLAS=H", "/*"]
}
```

The response from the POST method to create JCL to export the software instance specifies the members in the JCL data set that contains the generated jobs to perform the export action. You can use the z/OS jobs REST interface to submit and obtain the status of the export job. For example:

Submit the export job:

```
PUT /zosmf/restjobs/jobs HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
{
  "file": "'// 'USERID.SMJCL.CNTL(IZUD01EX)'"
}
```

The response to this PUT method provides the job ID and the job name for the submitted job.

Obtain status for the export job:

```
GET /zosmf/restjobs/jobs/IZUD01EX/job-id HTTP/1.1
Host: sys123.yourco.com
```

Sample REXX exec

A sample REXX exec that is named IZUDXEXP is provided in the SYS1.SAMPLIB data set to illustrate a program that uses Software management services to do the following:

- Add a software instance.
- Load the SMP/E Products, Features, and FMIDs for the software instance.
- Generate JCL to export the software instance.
- Run the generated JCL to create a portable software instance.

The sample REXX exec uses the HTTP REXX client of the z/OS Web Enablement Toolkit to perform HTTP operations to the z/OSMF server.

Modify the properties of a software instance

You can use this operation to modify the properties of a software instance that is defined in z/OSMF, including changing the global zone, target zones, or non-SMP/E managed data sets associated with the software instance. The modify operation updates only the definition of the software instance in z/OSMF. The physical data sets that compose the software instance are not affected.

HTTP method and URI path

```
PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

```
PUT /zosmf/swmgmt/swi/<swi-uuid>
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be modified. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be modified. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document included as input with this request.

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

Your request must include a JSON object that describes all the properties of the software instance to be modified, for example:

```

{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "datasets": [
    {
      "dsname": "data-set-name",
      "volume": "volume-serial"
    }
  ]
  "products": [
    {
      "prodname": "product-name",
      "prodid": "product-id",
      "release": "product-level",
      "vendor": "vendor-name",
      "url": "product-url",
      "features": ["feature-name"],
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date"
    }
  ],
  "workflows": [
    {
      "name": "workflow-name",
      "description": "workflow-description",
      "location": {
        "smptype": "smp-type",
        "smpname": "smp-name",
        "dsname": "workflow-dsname",
        "path": "workflow-path"
      },
      "performonhosts": true | false
    }
  ],
  "datasetproperties": [
    {
      "dddefname": "dddef-name",
      "zone": "zone-name",
      "dsname": "data-set-name",
      "volume": "volume-serial",
      "dstype": "DLIB",
      "properties": [{"key": "value"}]
    }
  ],
  "datasetpropertylabels": [
    {
      "propertyname": "property-name",
      "label": "property-label"
    }
  ],
  "productproperties": [
    {
      "prodid": "product-id",
      "release": "product-level",
      "prodname": "product-name",
      "properties": [{"key": "value"}]
    }
  ]
}

```

Figure 217. Modifying a software instance: request content

where:

swi-name

Name of the software instance. The name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ _ # @ ^). The name is required and must be unique on the system.

system-nickname

Nickname of the system that has access to the volumes and data sets where the software instance resides. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems defined to z/OSMF, you can use the z/OSMF topology services. For more details, see [“Topology services”](#) on page 687.

swi-description

Description of the software instance. The description is optional and can contain a maximum of 256 characters.

global-zone

CSI data set that contains the global zone that is used to manage the software. The data set name is required, must comply with z/OS data set naming conventions, and must end with .CSI.

target-zones

Comma-separated list of the target zones to be included in the software instance. At least one target zone is required. Each target zone name must be 1-7 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and special characters (@ # \$). The first character must be an alphabetic character.

categories

Comma-separated list of the categories to which the software instance is assigned. Each category name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ _ # @ ^). Assigning the software instance to a category is optional.

datasets

Array that contains each non-SMP/E managed data set to be added to the software instance. Adding non-SMP/E managed data sets to the software instance is optional.

data-set-name

Name of the non-SMP/E managed data set. You cannot specify data set members or a subset of a data set. A data set name is required if you are adding non-SMP/E managed data sets to the software instance. The data set name must comply with z/OS data set naming conventions

volume-serial

Volume on which the non-SMP/E managed data set resides. The volume serial is required if the data set is not cataloged, and the volume must be accessible by the system where the software instance resides. The volume serial must be 6 characters long; the valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9).

products

List of products for the software instance that are not managed by SMP/E. This list is optional.

product-name

Name of the product, but can be up to 64 characters.

product-ID

Identifier for the product, but can be up to 64 characters.

product-level

Release level for the product, but can be up to 64 characters.

vendor-name

Name of the vendor that provides the product, but can be up to 64 characters.

product-URL

A URL that links to additional information about the product, but can be up to 1023 characters.

feature-name

List of names of features for the product, but can be up to 64 characters.

general-availability-date

Date this level of the product is available to all users. May be null, or a date value, in ISO 8601 format, yyyy-mm-ddThh:mm:ssZ.

end-of-service-date

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

null

The end of service date is unknown for the product.

yyyy-mm-ddThh:mm:ssZ

The known end of service date, in ISO 8601 format.

NotAnnounced

The end of service date is not yet announced for the product.

workflows

List of workflows for the software instance. This list is optional.

workflow-name

Name for the workflow. The name may contain up to 100 characters, but must not include the characters for ampersand ('&'), forward slash ('/'), backward slash ('\'), logical or ('|'), greater than ('>'), or less than ('<'). Embedded blanks are allowed.

workflow-description

Description for the workflow. The description is optional, and can contain a maximum of 256 characters.

location

Location of the workflow definition file for the workflow. A workflow definition file location is required.

smp-type

The SMP/E element type for a workflow definition file that is managed by SMP/E. The SMP/E element type is optional and may be up to 12 uppercase alphanumeric characters. The first character cannot be numeric.

smp-name

The SMP/E element name for a workflow definition file that is managed by SMP/E. The SMP/E element name is optional and may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

workflow-dsname

The sequential data set name, or partitioned data set name and member for a workflow definition file that is in a data set. The data set name is optional, but if specified must comply with z/OS data set naming conventions. For example, IBMUSR.SM.WRKFLWS(WRKFLOW1).

workflow-path

The UNIX path for a workflow definition file that is a UNIX file. The UNIX path is optional, but if specified it must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash.
- Can be up to 1023 characters long.

performonhostsystm

Indicates whether the workflow steps may be performed on the host system or on another system. This property is optional and the default value is true.

true

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

false

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

Tip: Ensure that the description property contains information that helps the user select an appropriate alternative system for the performance of the workflow steps. For example, if the workflow steps are for installation verification procedures (IVPs), the user would select the system on which the newly activated software runs.

datasetproperties

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,” on page 1297](#) for more information.

To define properties for SMP/E managed data sets, specify the DDDEF entry name that identifies the desired data set or data sets. Specify the zone name only when necessary to identify a unique data set. For example, if there are more than one DDDEF entries with the same name pointing to different data sets, then both a DDDEF entry name and zone name are required to define properties for only one of those data sets. If an SMP/E zone name is not specified, then the properties apply to all data sets identified by all DDDEF entries with the same name in all zones that reside in the software instance.

To define properties for a non-SMP/E managed data set, specify the data set name and the volume for the data set as they are used when defining the data set to the software instance. That is, if only a data set name is specified to define the data set to the software instance, then specify only the data set name when defining properties for the data set. If a data set name and volume are specified to define the data set to the software instance, then specify both the data set name and volume to define properties for the data set.

dddefname

The name of the SMP/E DDDEF entry that describes one or more SMP/E managed data sets. It may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

zone

The zone name where the DDDEF entry resides. It may be up to 7 uppercase alphanumeric, \$, #, or @ characters.

dsname

The name of the subject data set. This is used to identify a non-SMP/E managed data set. It may be up to 44 characters and must comply with z/OS data set naming conventions.

volume

The volume of the subject data set. This is used to identify a non-SMP/E managed data set where the volume had been specified to identify an uncatalogued data set. It may be up to 6 uppercase alphanumeric, \$, #, or @ characters.

dstype

The usage type of the identified data set. The value may be DLIB or the default value of null. A value of DLIB indicates the identified data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone. The dstype value of DLIB must be specified for properties that identify distribution library or related control data sets so Software Management can safely ignore this property specification when a software instance does not contain SMP/E managed distribution libraries or related control data sets.

properties

A list of one or more properties for the subject data set, which is specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

datasetpropertylabels

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

propertyname

The name, or key, of the existing provider defined property.

label

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

productproperties

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 1297 for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

prodid

The identifier for the subject product. This is required to identify an SMP/E managed product. The product identifier may be up to 64 characters.

release

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product. The release level may be null, or up to 64 characters.

prodname

The name of the subject product. This is required to identify a non-SMP/E managed product. The product name may be null, or up to 64 characters.

properties

A list of one or more properties for the subject product, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

Usage considerations

The SMP/E global zone and target zones define the installed products, features, and FMIDs for a software instance. If you modify these properties for a software instance, perform a PUT method to load the products, features, and FMIDs for the updated software instance. Doing so refreshes the product, feature, and FMID information known to z/OSMF for the subject software instance. For instructions, see [“Load the products, features, and FMIDs for a software instance”](#) on page 495.

For other usage considerations, see [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

To submit a modify software instance request through the software management services, the user ID initiating the request requires the same authorizations as when performing a modify operation that uses the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task. The user ID must also have either CONTROL or UPDATE access to the SAF resources corresponding to the software instance to be modified, as follows:

- The name, system, and categories properties are used to create the SAF resource names for the software instance; therefore to modify any of these properties, the user ID must have CONTROL access to the existing SAF resource names and to the new SAF resource names.
- If a global zone or categories are implicitly added to z/OSMF through the modify operation, the user ID must have CONTROL access to the SAF resources corresponding to the specified global zone and categories.
- To modify any other property, the user ID must have UPDATE access.

For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 460.

Example

In the following example, the PUT method is used to modify the description property for software instance *Hooli* on system *PEV174*.

```

PUT /zosmf/swmgmt/swi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com Content-Type: application/json Accept-Language:
en { "name": "Hooli", "system": "PEV174", "description": "Fictitious software
product from Hooli", "globalzone": "IBMUSER.HOOL.I.CSI", "targetzones":
[{"TGTZ": "Hooli", "dsname": "IBMUSER.HOOL.I.PROCLIB", "volume": "LV1234"},
{"dsname": "IBMUSER.HOOL.I.PARMLIB", "volume": "LV1234"} ], "products": [ {"prodname": "Hooli
Express", "release": "2.1", "vendor": "Hooli Software"} ], "workflows": [ {"name": "Setup
Hooli", "description": "Setup and configure fictitious software product
Hooli", "location": {"dsname": "IBMUSER.HOOL.I.WORKFLOW(SETUP)"}, "datasetproperties":
[ {"dddefname": "HOOLMOD", "properties": [{"hooli-LnkLst": "yes"}, {"hooli-ispfType": "LMOD"}]},
{"dddefname": "HOOLMSG", "properties": [{"hooli-ispfType": "MESSAGE"}]},
{"dddefname": "HOOLPNL", "properties": [{"hooli-ispfType": "PANEL"}]},
{"dddefname": "HOOLSKL", "properties": [{"hooli-ispfType": "SKELETON"}]},
{"dddefname": "HOOLTLB", "properties": [{"hooli-ispfType": "TABLE"}]},
{"dddefname": "HOOLREXX", "properties": [{"hooli-ispfType": "EXEC"}]},
{"propertyname": "hooli-LnkLst", "label": "LNKLST Eligible"} ], "datasetpropertylabels":
{"propertyname": "hooli-ispfType", "label": "ISPF Element Type"} ] }

```

Figure 218. Sample request to modify a software instance

Figure 219 on page 495 provides a sample response, indicating that the update was successful.

```

HTTP/1.1 200 OK
Date: Tues, 22 July 2014 18:53:27 +0000GMT

```

Figure 219. Sample response for a modify software instance request

Load the products, features, and FMIDs for a software instance

You can use this operation to analyze the SMP/E global zone and target zones for a software instance to identify the installed products, features, and FMIDs in the instance, and to load that information into the z/OSMF Software Management task database.

HTTP method and URI path

```
PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/products
```

```
PUT /zosmf/swmgmt/swi/<swi-uuid>/products
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname>/<swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to retrieve product, features, and FMIDs. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).

- **products** indicates that the products, features, and FMIDs included in the software instance are to be obtained and loaded into the Software Management task database.

When you issue this request, z/OSMF searches the SMP/E global zone and target zones that are associated with the software instance and gathers information about the software instance and its products, features, and FMIDs. z/OSMF returns this information in a JSON object, and stores this information in the Software Management task database.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content that is provided by the caller. Use the JSON content type ("Content-Type: application/json") if a JSON document is included as input with this request.

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

If the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. To do so, include the following JSON object in your request:

```
{
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 220. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server

where:

zosmf-user-ID

User ID for authenticating with the secondary z/OSMF instance.

zosmf-password

Password for authenticating with the secondary z/OSMF instance.

proxy-user-ID

User ID for authenticating with the HTTP proxy server.

proxy-password

Password for authenticating with the HTTP proxy server.

Include the JSON object in the request only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation that uses the z/OSMF Software Management task. That is, to retrieve the product, feature, and FMID information, the user ID must have READ access to the Software Management task and UPDATE access to the SAF resources for the software instance that is being updated. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request was accepted. If the request was accepted, the service returns status code 202 Accepted and a JSON object that contains a URL (`{"statusurl": "url"}`). To monitor the status of the retrieve product, feature, and FMID information request and to obtain the results, perform GET requests to the supplied URL. Only the user ID that initiates the retrieve product, feature, and FMID information request is authorized to check the status and obtain the results. One of the following responses is returned from the get status request:

- If the retrieve product, feature, and FMID information request is still in progress, an HTTP response code of 200 OK is returned, along with the following JSON object: `{"status": "status"}`.
- If the retrieve product, feature, and FMID information request is complete, an HTTP response code of 200 OK is returned, along with the following JSON object:

```
{
  "status": "status",
  "swi": {
    "name": "swi-name",
    "system": "system-nickname",
    "description": "swi-description",
    "globalzone": "global-zone-name",
    "targetzones": ["target-zone-name"],
    "categories": ["category-name"],
    "productinfo": {
      "retrieved": "date-retrieved",
      "lastmodified": "date-modified",
      "modifiedby": "modified-user-ID",
      "created": "date-created",
      "createdby": "created-user-ID",
      "locked": "date-locked",
      "lockedby": "locked-user-ID",
      "datasets": [
        {
          "dsname": "data-set-name", "volume": "volume-serial"
        }
      ]
    },
    "products": [
      {
        "prodname": "product-name",
        "prodid": "product-ID",
        "release": "version-release-modification",
        "vendor": "vendor-name",
        "generalavailability": "general-availability-date",
        "endofservice": "end-of-service-date",
        "url": "product-URL",
        "productinfo": {
          "fileversion": "product-file-version",
          "features": [
            {
              "feature": "feature-name",
              "fmids": [
                {
                  "fmid": "fmid-name",
                  "description": "fmid-description",
                  "targetzones": ["fmid-target-zone-name"]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Where:

status

Status of the retrieve product, feature, and FMID information request. The status is either *running* or *complete*.

swi

JSON object to describe a software instance.

swi-name

Name of the software instance.

system-nickname

Nickname of the z/OSMF host system that has access to the global zone CSI data set included in the software instance. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

swi-description

Description of the software instance.

global-zone-name

Name of the CSI data set that contains the global zone that is used to manage the software.

target-zone-name

Array of the target zones in the specified global zone that describe the software.

category-name

Array of the categories to which the software instance is assigned.

date-retrieved

Date and time in ISO 8601 format that the product, feature, and FMID information for the software instance was last retrieved from the CSI data set. For example, 2014-08-20T19:23:25+00:00Z.

date-modified

Date and time in ISO 8601 format that the software instance was last modified. For example, 2014-08-20T19:23:25+00:00Z.

modified-user-ID

User ID of the user who last modified the software instance.

date-created

Date and time in ISO 8601 format that the software instance was created. For example, 2014-08-20T19:23:25+00:00Z.

created-user-ID

User ID of the user who created the software instance.

date-locked

Date and time in ISO 8601 format that the software instance was locked for an impending update. For example, 2014-08-20T19:23:25+00:00Z. If null, the software instance is not locked.

locked-user-ID

User ID of the user who locked the software instance.

datasets

Array of the non-SMP/E managed data sets that are included in the software instance.

data-set-name

Name of the data set.

volume-serial

Volume serial for the volume where the data set resides.

products

Array of the products included in the software instance.

product-name

Name of the product.

product-ID

Identifier for the product.

version-release-modification

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

vendor-name

Name of the vendor that provides the product.

general-availability-date

Date and time in ISO 8601 format that a version or release of the product became available to all users. For example, 2014-08-20T19:23:25+00:00Z.

end-of-service-date

Date and time in ISO 8601 format that service support ends for the product. For example, 2014-08-20T19:23:25+00:00Z.

product-URL

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions.

product-file-version

Version of the most recent product information file that supplied information about the product. The version represents the date and time in ISO 8601 format that file was created or last updated. For example, 2014-08-20T19:23:25+00:00Z.

features

Array of the features contained in the software instance.

feature-name

Name of the feature.

fmids

Array of the FMIDs contained in the software instance.

fmid-name

Name of the FMID.

fmid-description

Description of the FMID.

fmid-target-zone-name

Array of the target zones where the FMID is installed.

- If the retrieve product, feature, and FMID information request is complete but the results are no longer available, an HTTP response code of 404 Not Found is returned. z/OSMF makes the results available for a client application for a finite period of time. When that time elapses, the results are no longer available; in which case, the client must reissue the request.

If the retrieve product, feature, and FMID information request cannot be processed, a status code of 4nn or 5nn is returned, indicating that an error has occurred. For more details, see [“Error handling” on page 460](#).

Example

In the following example, the PUT method is used to retrieve the product, feature, and FMID information for software instance *DB2V9* on system *SYS123*.

```
PUT /zosmf/swmgmt/swi/SYS123/DB2V9/products HTTP/1.1
Host: sys123.yourco.com
```

Figure 221. Sample request to retrieve the product, feature, and FMID information for a software instance

[Figure 222 on page 500](#) provides a sample response, indicating that the retrieve product, feature, and FMID information request has been accepted and supplying the URL to use for monitoring the status of that request.

```

HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgt/statusmonitor/prodload
/4837290198343"}

```

Figure 222. Sample response for a retrieve product, feature, and FMID information request

To check the status of the retrieve product, feature, and FMID information request, submit the following request:

```

GET /zosmf/swmgt/statusmonitor/prodload/4837290198343 HTTP/1.1
Host: sys123.yourco.com

```

Figure 223. Sample request to obtain the status of a retrieve product, feature, and FMID information request

Figure 224 on page 500 provides a sample get status response, indicating that the retrieve product, feature, and FMID information request is in progress.

```

HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running"}

```

Figure 224. Sample get status response when the retrieve product, feature, and FMID information request is in progress

Figure 225 on page 500 provides a sample get status response, indicating that the retrieve product, feature, and FMID information request is complete.

```

HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:36 +00006GMT
Content-Type: application/json
Content-Language: en
Connection: close
{
  "status":"complete", "swi":{"
    "name":"DB2V9", "system":"PEV174", "description":null,
    "globalzone":"DB2.GLOBAL.CSI", "targetzones":["DB2TGT"], "categories":null,
    "productinfo":{"retrieved":"2014-08-20T19:23:25Z", "lastmodified":"2014-08-
    20T19:23:25Z", "modifiedby":"FRED", "created":"2014-08-20T19:23:25Z",
    "createdby":"BARNEY", "locked":null, "lockedby":null, "datasets":
    [{"dsname":"USER.DB2V9.PROCLIB", "volume":"LV1234"},
    {"dsname":"USER.DB2V9.SAMPLES", "volume":"LV1234"}]}, "products":
    [{"prodname":"DB2 for z/OS", "prodid":"5635-DB2", "release":"09.01.00",
    "vendor":"IBM", "generalavailability":"20006-06-09T19:23:25Z",
    "endofservice":"2014-06-27T19:23:25Z", "url":null,
    "productinfofileversion":"2014-01-01", "features":[{"feature":"DB2 Base",
    "fmids":[{"fmid":"HDB9910", "description":"DB2 BASE/TSO", "targetzones":
    ["DB2V9T"]}]}]}]}
  }
}

```

Figure 225. Sample get status response when the retrieve product, feature, and FMID information request is complete

Delete a software instance

You can use this operation to remove a software instance definition from z/OSMF. The delete operation removes only the definition of the software instance from z/OSMF. The physical data sets that compose the software instance are not affected.

HTTP method and URI path

```
DELETE /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

```
DELETE /zosmf/swmgmt/swi/<swi-uuid>
```

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be deleted. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be deleted. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

To submit a delete software instance request through the software management services, the user ID initiating the request requires the same authorizations as when performing a remove operation that uses

the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task, and CONTROL access to the SAF resources corresponding to the software instance to be deleted. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error has occurred. For more details, see [“Error handling” on page 460](#).

Example

In the following example, the DELETE method is used to delete software instance *DB2V9* on system *PEV174*.

```
DELETE /zosmf/swmgmt/swi/PEV174/DB2V9 HTTP/1.1
Host: pev174.yourco.com
```

Figure 226. Sample request to delete a software instance

[Figure 227 on page 502](#) provides a sample response, indicating that the delete operation was successful.

```
HTTP/1.1 200 OK
Date: Tues, 22 July 2014 18:53:27 +00006GMT
```

Figure 227. Sample response for a delete software instance request

Deleting the Temporary Catalog Aliases

When a software instance is created by the deployment action, a new master catalog may also be created. Temporary catalog aliases and data set aliases are created during the deployment to allow access to the target software instance data sets from both the driving and target systems. The "delete temporary catalog alias" action removes the temporary catalog aliases and data set aliases.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/deltempcatalias
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/deltempcatalias
```

Where:

- *zosmf/swmgmt* identifies the software management services.
- *swi* informs the service that the request is for the software instance object.
- *<system-nickname>/<swi-name>* further qualifies the request and indicates the specific software instance to be processed. A software instance is uniquely identified by its name *swi-name* and the nickname *system-nickname* of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be processed. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF”](#) on [page 461](#).
- */deltempcatalog* indicates that the temporary catalog aliases and data set aliases that are created during the deployment process are to be deleted.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type `Content-Type: application/json` is used for the JSON document, if there are any, and included as an input with this request.

Custom headers

None.

Request content

The request must include a JSON document that identifies properties that are required to perform the operation.

```
{
  "jcldataset": "data-set-name",
  "jobstatement": ["jclrecord"],
  "tgtssystem": "system-nickname",
  "tgttjobstatement": ["jclrecord"],
  "unixdatasets": [{
    "dsname": "data-set-name",
    "mountpoint": "UNIX-path"
  }],
  "systems": [{
    "name": "system-nickname",
    "zosmfuid": "user-id",
    "zosmfpw": "password",
    "proxyuid": "user-id",
    "proxypw": "password"
  }]
}
```

Where:

jcldataset

Partitioned data set to contain the generated JCL. The data set name must comply with z/OS® data set naming conventions. If a data set with the specified name does not already exist, z/OSMF will create a data set residing on the target system for the software instance.

tgtsystem

Nickname of the system that is currently started with, or will be started with, the new master catalog created during the deployment for the subject software instance. This system has or will have access to the software instance data sets after the temporary catalog aliases are deleted and may or may not be the same as the current system for the software instance.

Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems that are defined to z/OSMF, you can use the z/OSMF topology services. For more details, see Topology services.

jobstatement

List of JCL cards for the JOB statement to be used in the generated JCL that will run on the current system for the software instance. This is an optional property. If specified it must be a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be // or /* and the job name must be 1 - 8 characters. If the JOB statement is not provided, the default is exactly //IZUDnn^{tt} JOB (ACCOUNT), 'NAME' where nn^{tt} uniquely identifies the job sequence and type.

tgtjobstatement

List of JCL cards for the JOB statement to be used in the generated JCL that will run on the target system for the software instance. This is an optional property. If specified it must be a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/" and the job name must be 1 to 8 characters. If the JOB statement is not provided, the default is exactly //IZUDnn^{tt} JOB (ACCOUNT), 'NAME' where nn^{tt} uniquely identifies the job sequence and type.

unixdatasets

List of UNIX file system data sets in the software instance that are currently not mounted. Because they are currently not mounted, they cannot be identified by z/OSMF by referencing the UNIX directories that are defined in the SMP/E target zone DDDEF entries.

This is an optional property. Required only if the software instance describes SMP/E managed software, and if any of the UNIX file system data sets that contain that software are currently not mounted.

dsname

Name of a UNIX file system data set.

The data set name must comply with z/OS data set naming conventions.

mountpoint

Mount point for the UNIX file system data set, if the data set were mounted.

Must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash, unless a slash is the only character. That is, "/" (root) is valid, but "/mountpoint/" is not.
- Can be up to 1023 characters long.

systems

List of authentication credentials for remote z/OSMF servers. This is an optional property.

system

Nickname of the system for a remote z/OSMF server. This name must match either the system where the software instance resides or the target system for the operation.

zosmfuid

Userid for authenticating with the named remote z/OSMF server. This is an optional property.

zosmfpw

Password for authenticating with the named remote z/OSMF server. This is an optional property.

proxyuid

Userid for authenticating with an HTTP proxy. This is an optional property.

proxypw

Password for authenticating with an HTTP proxy. This is an optional property.

The request content is required, but some properties are optional. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to

authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. Therefore, you may need to specify the remote z/OSMF user ID, password, and proxy user ID and password on the systems property.

Required authorizations

To submit requests through the software management services, the user ID that initiates the request requires the same authorizations as when you perform an analogous operation that uses the z/OSMF Software Management task. That is, to delete the temporary catalog aliases for a software instance, the user ID must have READ access to the Software Management task, and UPDATE access to the SAF resources for the software instance that is being processed. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

Generating JCL to delete the temporary catalog aliases for a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document that contains a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example:

```
{
  "statusurl": "url"
}
```

statusurl

Indicates the URL that provides status for the "delete temporary catalog aliases" request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document that contains status information for the operation.
- If the operation is complete, then an HTTP response code of 200 OK is returned, along with a JSON document that contains status information and the desired result set.
- If the request is expired, then an HTTP response code of 404 Not found is returned. That is, when the operation is complete, the z/OSMF server holds the result set for a finite length of time. After that length of time has passed, the result set is said to expire and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",
  "percentcomplete": "percent",
  "jcl": ["data-set-name(member-name)"],
  "jobs": [{
    "jobname": "job-name",
    "jobid": "job-id",
    "retcode": "completion-code"
  }],
  "swiurl": "swi-URL"
}
```

status

The status of the "delete temporary catalog aliases" request. The status can be either "running" or "complete".

percentcomplete

The percentage of the processing that is complete for this delete temporary catalog aliases request, expressed as a whole number from 0 to100.

jcl

Ordered list of generated jobs for the delete temporary catalog aliases action. Each job is a unique member in the JCL data set. The values are fully qualified data set and member names of the form "data-set-name(member-name)".

jobs

Ordered list of batch jobs submitted to delete the temporary catalog aliases.

jobname

Job name.

jobid

Job ID.

retcode

Job completion code.

swiurl

URL that allows you to access the updated software instance. For example, a client application can use the URL to get the software instance properties or list the data sets in the software instance.

For more information about the error response document that contains a reason code and a list of one or more message strings to describe the errors that are detected during the processing of a request, see [“Error handling” on page 460](#).

Example

The example request [“#unique_16/unique_16_Connect_42_ExampleRequestRemoveTempCatAlias” on page 506](#) deletes the temporary catalog aliases for the software instance that is named zos25 on SYS123.

```
POST /zosmf/swmgmt/swi/SYS123/zos25/deltempcatalias HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "jcldataset": "USERID.SMJCL.CNTL",
  "jobstatement": ["//JOBNAME JOB (12345), 'USER', NOTIFY=&SYSUID",
  "// MSGCLASS=H, MSGLEVEL=1, REGION=0M", "/*"],
  "tgtssystem": "SYSNEW"
}
```

A sample response is described in [“#unique_16/unique_16_Connect_42_ExampleResponseRemoveTempCatAlias” on page 506](#):

```
HTTP/1.1 202 Accepted
Date: Tues, 1 February 2022 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl": "https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/deltempcatalias/4837290198343"}
```

The example response [“#unique_16/unique_16_Connect_42_ExampleResponseRemoveTempCatAlias” on page 506](#) indicates the request to delete temporary catalog aliases for the software instance is accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is described in [“#unique_16/unique_16_Connect_42_SampleGETrequestStatusMonitorURL” on page 506](#):

```
GET /zosmf/swmgmt/statusmonitor/deltempcatalias/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```


A sample response is described in “[#unique_16/unique_16_Connect_42_SampleGETresponseStatusMonitorURL](#)” on page 507:

```
HTTP/1.1 200 OK
Date: Tues, 1 February 2022 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status": "complete", "percentcomplete": "100",
"jcl": ["USERID.SMJCL.CNTL(IZUD01UC)",
"USERID.SMJCL.CNTL(IZUD01DA)",
"USERID.SMJCL.CNTL(IZUD01DT)"],
"jobs": [{"jobname": "IZUD01UC", "jobid": "JOB12345", "retcode": "CC 0"},
{"jobname": "IZUD02DA", "jobid": "JOB12347", "retcode": "CC 0"},
{"jobname": "IZUD03DT", "jobid": "JOB12349", "retcode": "CC 0"}],
"swiurl": "https://sys123.yourco.com/zosmf/swmgt/swi/SYSNEW/zos25"
```

List the portable software instances defined to z/OSMF

You can use this operation to obtain a list of the portable software instances that are defined to a z/OSMF instance.

HTTP method and URI path

```
GET /zosmf/swmgt/pswi
```

where:

zosmf/swmgt

Identifies the software management services.

pswi

Informs the service that the request is for a portable software instance object.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

None.

Usage considerations

See “[Usage considerations for the z/OSMF REST services](#)” on page 4.

Required authorizations

Certain authorizations are required to submit a request through the software management services to list portable software instances. The user ID that initiates the request requires the same authorizations as

the ones needed to perform a list operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information about errors, see [“Error handling” on page 460](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "pswilist": [
    {
      "name": "pswi-name",
      "system": "system-nickname",
      "description": "pswi-description",
      "directory": "UNIX-directory",
      "categories": ["categories"],
      "lastmodified": "last-modified",
      "modifiedby": "modified-user-ID",
      "created": "date-created",
      "createdby": "created-user-ID",
      "locked": "date-locked",
      "lockedby": "locked-user-ID"
    }
  ]
}
```

where:

pswilist

Array that contains each portable software instance that is defined to z/OSMF.

pswi-name

Name of the portable software instance.

system-nickname

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 687](#).

pswi-description

Description of the portable software instance.

UNIX-directory

UNIX directory that contains the portable software instance files.

categories

Comma-separated list of the categories to which the portable software instance is assigned.

last-modified

Date and time, in ISO 8601 format, that the portable software instance was last modified.

modified-user-ID

User ID of the user who last modified the portable software instance.

date-created

Date and time, in ISO 8601 format, that the portable software instance was created.

created-user-ID

User ID of the user who created the portable software instance.

date-locked

Date and time, in ISO 8601 format, that the portable software instance was locked. This attribute is null if the portable software instance is not currently locked.

locked-user-ID

User ID of the user who locked the portable software instance. This attribute is null if the portable software instance is not currently locked.

Example

In the following example, the GET method is used to retrieve a list of the portable software instances that are defined on the z/OSMF host system zosmf1.yourco.com.

```
GET /zosmf/swmgmt/pswi HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 228. Sample request to retrieve a list of portable software instances

A sample response is shown in Figure 229 on page 509.

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close

{"pswilst":[{"name":"DB2V9", "system":"PEV174", "description":null,
"directory":"/u/zosmfuser/DB2V9","categories":null,
"lastmodified":"2020-06-01T19:23:25+00:00", "modifiedby":"FRED",
"created":"2020-01-03T12:00:10+00:00", "createdby":"BARNEY", "locked":null,
"lockedby":null}]}
```

Figure 229. Sample response from a request to retrieve a list of portable software Required authorizations

Deploy a software instance

You can use the POST method to perform a deployment operation which will create a new software instance.

Currently this method has the following restrictions and limitations:

- You can deploy a software instance only, not a portable software instance.
- Target software instance data set names will exactly match the source software instance data set names. You cannot change data set names.
- A new master catalog for the target software instance data sets will be created unconditionally.

HTTP method and URI path

```
POST /zosmf/swmgmt/dep
```

where:

zosmf/swmgmt

Identifies the software management services.

dep

Informs the service that the request is for a deployment.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content that is provided by the caller. The following value is acceptable:

- Content-Type: application/json (JSON)

The JSON content type is used for the JSON document that is included as input with this request.

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

```
{
  "name": "deployment-name",
  "description": "deployment-description",
  "source": {
    "type": "swi",
    "name": "name",
    "system": "system-nickname"
  },
  "targetvolumes": [ {
    "volume": "volume-serial",
    "indirectcatalogsymbol": "volume-symbol"
  } ],
  "mastercatalog": {
    "name": "catalog-name",
    "volume": "volume-serial",
    "tempcatalias": "temporary-catalog-alias"
  },
  "jobstatement": [ "jcl-record" ],
  "jcldataset": "data-set-name",
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

where:

name

The name for the Deployment object to be created in the SM db.

description

The description for the Deployment object. This value is optional.

source

Identifies the source software for the deployment operation. A copy of the specified software instance is the result of the deployment operation.

type

The type of source object for the deployment: software instance ("swi").

name

The name for the source software instance.

system

Nickname of the z/OSMF host system that has access to the volumes and data sets where the source software instance is. The z/OSMF host system is where the target software instance data sets are created. Use the nickname for the system definition in the z/OSMF Systems task.

targetvolumes

A list of target volumes. The target software instance data sets are created on these volumes. The number of input target volumes must be equal to or greater than the number of volumes on which the source software instance data sets reside.

name

The volume serial for a target volume.

indirectcatalogsymbol

If specified, this value indicates that target data sets are indirectly cataloged and is the symbol for the volume serial in the data set catalog entries. For example, "&SYSR1" or "*****". This value is optional. The default behavior is to directly catalog all target data sets.

mastercatalog

Identifies the new master catalog to be created. All target software instance data sets are cataloged in this new catalog. The new catalog is an optional property. If not specified, the target software instance data sets are cataloged in the existing driving system catalog.

name

Name for the new master catalog.

volume

Volume serial where the new master catalog is created.

tempcatalias

The temporary catalog alias used to uniquely identify the data sets in the subject catalog from the z/OS driving system catalog. 1 - 8 characters and must follow data set name syntax.

jobstatement

List of JCL cards for the JOB statement to be used in the generated JCL for the deployment operation. The jobstatement is an optional property. If specified, it must be a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 - 8 characters. If no JOB statement is provided, the default is exactly "/IZUDnntt JOB (ACCOUNT),'NAME'" where "nntt" uniquely identifies the job sequence and type.

jcldataset

Partitioned data set to contain the generated deployment JCL. The data set name must comply with z/OS data set naming conventions.

zosmfuid

User ID for authenticating with a remote z/OSMF instance. This is an optional property.

zosmfpw

Password for authenticating with a remote z/OSMF instance. This is an optional property.

proxyuid

User ID for authenticating with an HTTP proxy. This is an optional property.

proxypw

Password for authenticating with an HTTP proxy. This is an optional property.

The request content is required, but some properties are optional. For example, if the software instance is not in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance. This secondary z/OSMF instance is running in the sysplex where the software instance is. If the primary z/OSMF instance must use an HTTP proxy server to connect with the secondary z/OSMF instance, you might need to authenticate with that HTTP proxy server. Therefore, you may need to specify the remote z/OSMF user ID, password, and proxy user ID and password.

Required authorizations

Certain authorizations are required to submit a request through the software management services to deploy a software instance. The user ID that initiates the request requires the same authorizations as the user IDs that are needed to perform a deployment operation with the z/OSMF Software Management task.

The user ID needs the following:

- READ access to the Software Management task.
- READ access to the SAF resources that correspond to the source software instance.
- CONTROL access to the SAF resources that correspond to the deployment object to be created.
- CONTROL access to the SAF resources that correspond to the target software instance to be created.

For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*

Expected response

Deploying a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted. It also returns a JSON document that contains a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set.

```
{
  "statusurl": "url"
}
```

where:

statusurl

Indicates the URL that provides status for the deployment request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document that contains status information for the operation.
- If the operation is complete, then an HTTP response code of 200 OK is returned, along with a JSON document that contains status information and the desired result set.
- If the request expires, then an HTTP response code of 404 Not found is returned. That is, when the operation is complete, the z/OSMF server holds the result set for a finite length of time. After that time passes, the result set is said to expire and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",

```

```

"percentcomplete":"percent",
"tgtswiname":"name",
"tgtswiurl":"swi-URL",
"jcl":["data-set-name(member-name)"],
"jobs":[{
  "jobname":"job-name",
  "jobid":"job-id",
  "retcode":"completion-code"
}],
"tgtswidatasets":[{
  "aliases":["alias"],
  "catalog":"catalog-name",
  "blksize":"block-size",
  "dddefs":[{
    "dddef":"dddef-name",
    "zone":"zone-name",
    "path":"unix-directory"
  }],
  "dscategory":["data-set-category"],
  "dsname":"data-set-name",
  "dstype":"data-set-type",
  "extents":"allocated-extents",
  "extendedformat":true | false,
  "lrecl":"logical-record-length",
  "mountpoint":"unix-path",
  "unixdirs":["unix-directory"],
  "recfm":"record-format",
  "storclas":"storage-class",
  "tempcatalias":"temporary-catalog-alias",
  "tracks":"allocated-tracks",
  "used":"used-tracks-percent",
  "volumes":["volume-serial"]
}]
}

```

where:

status

The status of the deployment request. The status can be either "running" or "complete".

percentcomplete

The percentage of the processing that is complete for this deployment request, expressed as a whole number 0 - 100.

tgtswiname

Name for the target software instance created by the deployment operation. z/OSMF generates a name for the target software instance from the name of the source software instance, by appending a sequence number, like this: source-name_n where "n" is an integer. For example, if the source software instance name is zOS25, then the generated target software instance name is zOSV25_1. If a software instance with the generated name exists, then the sequence number is incremented until a unique name is generated. For example, zOSV25_2, zOSV25_3.

tgtswiurl

URL that allows you to access the target software instance. For example, a client application can use the URL to read a software instance or list the data sets in the target software instance.

jcl

Ordered list of generated JCL (jobs) for the deployment operation. Each job is a unique member in the JCL data set. The values are fully qualified data set and member names of the form "data-set-name(member-name)".

jobs

Ordered list of batch jobs submitted for the deployment operation.

jobname

Job name.

jobid

Job ID.

retcode

Job completion code.

tgtswidatasets

The list of data sets in the target software instance created by the deployment operation.

aliases

A list of alias names for the data set.

catalog

For cataloged data sets, the name of the catalog that includes the data set.

blksize

The data set block size.

dddefs

The list of SMP/E zones and DDDEF entries that reference the data set.

dddef

The name of the DDDEF entry.

zone

The name of an SMP/E zone that contains the DDDEF entry.

path

The UNIX directory identified in the DDDEF entry.

dscategory

List of categories for how the data set is used. Can be one or more of the following:

- DLIB — SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL — SMP/E control data set associated with the global zone.
- SMP — SMP/E control data set.
- SMPTLIB — SMPTLIB data sets associated with the global zone.
- TARGET — SMP/E managed target library, or SMP/E control data set associated with a target zone.

- WORKFLOW — Contains one or more workflow definition files for the workflows that are explicitly defined to the software instance.
- OTHER — None of the above.

dsname

The data set name.

dstype

The type for the data set. Can be one of the following types:

- HFS — Hierarchical file system.
- PDS — Partitioned data set.
- PDSE — Partitioned data set extended.
- SEQ — Sequential data set.
- VSAM — VSAM data set.
- ZFS — zSeries file system.

extents

The number of extents that are allocated to the data set.

extendedformat

Indicates, true or false, if the data set is an extended format sequential data set.

lrecl

Data set logical record length.

mountpoint

Indicates the mount point for the UNIX file system data set.

unixdirs

A list of UNIX directories that are in the subject data set and contain one or more workflow definition files for the software instance.

recfm

Data set record format.

storclas

The name of the storage class where the data set resides.

tempcatalias

When you create an alternative master catalog, tempcatalias is the temporary catalog alias that is used to uniquely identify the data set from the z/OS driving system catalog. This value is the temporary catalog alias that is used for the alternate master catalog and is a data set name prefix to reference the data set from the driving system. To reference the data set from the driving system catalog, use a name that is constructed like this: tempcatalias.dsname.

The tempcatalias is null if a data set is cataloged in the new alternate master catalog indirectly, which means it uses a volume symbol instead of a specific volume serial. The tempcatalias is also null if it is not cataloged at all. Such data sets cannot be referenced by name from the driving system catalog. They must be referenced by using the volume instead.

If you are not creating an alternate master catalog, then temptations is null.

tracks

The number of 3390-device equivalent tracks (56664 bytes/track) allocated to the data set.

used

The percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

volumes

The list of volume serials where the data set resides.

Example

In the following example, the POST method is used to create a new software instance by deploying software instance zos24 on system SYS123.

```
POST /zosmf/swmgmt/dep HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "name": "zos24Deploy",
  "source": {"type": "swi", "name": "zos24"},
  "system": "SYS123",
  "targetvolumes": [{"volume": "LT1234"}, {"volume": "LT5678"}],
  "mastercatalog": {"name": "CATALOG.ZOS24.MCAT", "volume": "LT12345"},
  "tempcatalias": "ZOS24"},
  "jobstatement": ["//JOBNAME    JOB (123456), 'USER', NOTIFY=&SYSUID, REGION=0M, ",
  "// MSGCLAS=H", "/*"]
}
```

Figure 230. Sample request to retrieve a list of portable software instances

The response indicates that the deployment request is accepted and provides the URL to use for monitoring the status.

```
HTTP/1.1 202 Accepted
Date: Tues, 21 April 2021 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl": "https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/deployment/4837290198343"}
```

To check the status of the list deployment request, submit the following request:

```
GET /zosmf/swmgmt/statusmonitor/deployment/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response:

```
HTTP/1.1 200 OK
Date: Tues, 21 April 2021 18:53:19 +00005GMT
Content-Type: application/json
```

```
Content-Language: en
Connection: close
{"status":"running", "percentcomplete":"65"}
```

The response indicates the operation to deploy the software instance is still running and 65% complete. A final request to the status monitor URL is as follows:

```
GET /zosmf/swmgt/statusmonitor/deployment/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response is as follows:

```
HTTP/1.1 200 OK
Date: Tues, 21 April 2021 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "percentcomplete":"100",
"swiname":"zos24_1",
"swiurl":"https://zosmf1.yourco.com/zosmf/swmgt/swi/SYS123/zos24_1",
"jcl":["IBMUSR.SMJCL.CNTL(IZUD01CP)",
"IBMUSR.SMJCL.CNTL(IZUD02RN)",
"IBMUSR.SMJCL.CNTL(IZUD03UC)"],
"jobs":[{"jobname":"IZUD01CP","jobid":"JOB12345","retcode":"CC 0"},
{"jobname":"IZUD02RN","jobid":"JOB12347","retcode":"CC 0"},
{"jobname":"IZUD03UC","jobid":"JOB12349","retcode":"CC 0"}],
"tgtswidatasets": [
  {
    "aliases": [
      "ZOS24.IBMUSR.SWI.AMACLIB"
    ],
    "catalog": "CATALOG.ZOS24.MCAT",
    "blksize": "32720",
    "dddefs": [
      {
        "dddef": "AMACLIB",
        "zone": "DLIB"
      },
      {
        "dddef": "AMACLIB",
        "zone": "TGT"
      }
    ],
    "dscategory": [
      "DLIB"
    ],
    "dsname": "IBMUSR.SWI.AMACLIB",
    "dstype": "PDSE",
    "extents": "1",
    "extendedformat": false,
    "lrecl": "80",
    "mountpoint": null,
    "unixdirs": null,
    "recfm": "FB",
    "storclas": null,
    "tempcatalias": "ZOS24",
    "tracks": "1",
    "used": "50",
    "volumes": [
      "LT1234"
    ]
  }
]
```

```

    },
    {
      "aliases": [
        "ZOS24.IBMUSR.SWI.CSI"
      ],
      "catalog": "CATALOG.ZOS24.MCAT",
      "blksize": null,
      "dddefs": null,
      "dscategory": [
        "DLIB",
        "GLOBAL",
        "SMP"
      ],
      "dsname": "IBMUSR.SWI.CSI",
      "dstype": "VSAM",
      "extents": null,
      "extendedformat": false,
      "lrecl": null,
      "mountpoint": null,
      "unixdirs": null,
      "recfm": null,
      "storclas": null,
      "tempcatalog": "ZOS24",
      "tracks": "76",
      "used": null,
      "volumes": [
        "LT1234"
      ]
    },
    { ... list of remaining data sets ...}
  ]
}

```

Processing

The Software Management Deployment REST API does the following:

1. Validate the request content is of proper form and syntax.
2. Verify that the source software instance exists.
3. Create a new deployment using the name provided in the input. No categories are assigned to the deployment.
4. Analyze the source software instance to get the list of data sets and determine the number of volumes on which the source data sets reside.
5. Ensure that the number of input target volumes is equal to or greater than the number of source volumes.
6. Create the deployment configuration:
 - a. The zone names in the target software instance will match the zone names in the source software instance.
 - b. The target data set names will match the source data set names.
 - c. Target volumes are selected from the list of input target volumes. Each source volume corresponds to one target volume.
 - d. The new master catalog name, volume, and temporary catalog alias are provided in the input. All target data sets will be cataloged in this new master catalog.
7. Verify the target software instance described by the configuration can be created; make sure there are no unintended data set collisions on target volumes, and ensure the temporary catalog alias is not already defined in the driving system catalog.
8. Generate the deployment JCL.
9. Submit the deployment JCL, in sequence, one job at a time, ensuring an acceptable condition code before the next job is submitted. Exception: The RACF job is NOT submitted. It is overridden complete.
10. After all jobs have run successfully, the target software instance is created using a name generated from the source software instance name with an appended sequence number to ensure uniqueness.

Categories from the source software instance, if any, are also added to the target software instance definition.

11. Get the list of target software instance data sets and prepare the response content.

Delete a Deployment

You can use this operation to remove a completed Deployment object from z/OSMF. The delete operation removes only the definition of the deployment from z/OSMF.

The following are not affected:

- JCL data set containing the generated deployment jobs.
- Submitted deployment jobs and their output in the target system spool.
- Target software instance created by the deployment operation.
- Workflow instances created for the target software instance.

HTTP method and URI path

```
DELETE /zosmf/swmgmt/dep/<dep-name>
```

where:

zosmf/swmgmt

Identifies the software management services.

dep

Informs the service the request is for a deployment.

<dep-name>

Name for the deployment to be deleted.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

None.

Required authorizations

Certain authorizations are required to submit a request through the software management services to delete a deployment. The user ID that initiates the request requires the same authorizations as those needed to delete the deployment with the z/OSMF Software Management task. The user ID needs the following:

- READ access to the Software Management task
- CONTROL access to the SAF resources that correspond to the deployment object to be deleted.

For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more information about errors, see [“Error handling” on page 460](#).

Example

In the following example, the DELETE method is used to delete deployment Hooli from system PEV174.

```
DELETE /zosmf/swmgmt/dep/Hooli HTTP/1.1
Host: pev174.yourco.com
```

Here is the sample response that indicates that the delete operation was successful.

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2021 18:53:27 +00004GMT
```

Retrieve the properties of a portable software instance

You can use this operation to retrieve the properties of a portable software instance.

HTTP method and URI path

```
GET /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
```

where:

zosmf/swmgmt

Identifies the software management services.

pswi

Informs the service that the request is for a portable software instance object.

<system-nickname>

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located. In combination with *<pswi-name>*, it further qualifies the request and indicates the specific portable software instance to retrieve.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 687](#).

<pswi-name>

Name of the portable software instance. In combination with *<system-nickname>*, it further qualifies the request and indicates the specific portable software instance to retrieve.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

Certain authorizations are required to submit a request through the software management services to read a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform a read operation with the z/OSMF Software Management task. The user ID needs READ access to both the Software Management task and the SAF resources that correspond to the portable software instance to retrieve. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information about errors, see [“Error handling” on page 460](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "name": "pswi-name",
  "system": "system-nickname",
  "description": "pswi-description",
  "directory": "UNIX-directory",
  "categories": ["categories"],
  "serverxml": ["remote-server-xml"],
  "clientxml": ["client-options-xml"],
  "jcldataset": "jcl-data-set-name",
  "lastmodified": "last-modified",
  "modifiedby": "modified-user-ID",
  "created": "date-created",
  "createdby": "created-user-ID",
  "locked": "date-locked",
  "lockedby": "locked-user-ID",
  "signatureverificationkeyring": "keyring",
  "packagesigner certdetails": {
    "subjectname": "subject-name",
    "serialnumber": "serial-number",
    "fingerprint": "finger-print",
    "issuer": "issuer-subject-name"
  },
  "products": [{
    "prodname": "product-name",
    "productid": "product-ID",
    "release": "release-level",
    "vendor": "vendor-name",
    "generalavailability": "general-availability-date",
```

```

        "endofservice": "end-of-service",
        "url": "product-URL",
        "productinfofileversion": "file-version",
        "features": [
            "feature-name"
        ]
    },
    "nonsmpeproducts": [{
        "prodname": "product-name",
        "productid": "product-id",
        "release": "release-level",
        "vendor": "vendor-name",
        "generalavailability": "general-availability-date",
        "endofservice": "end-of-service-date",
        "url": "product-url",
        "productinfofileversion": "file-version",
        "features": [
            "feature-name"
        ]
    }]
}

```

where:

pswi-name

Name of the portable software instance.

system-nickname

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 687](#).

pswi-description

Description of the portable software instance.

UNIX-directory

UNIX directory that contains the portable software instance files.

categories

Comma-separated list of the categories to which the portable software instance is assigned.

remote-server-xml

Information from when the Add Portable Software Instance from Download Server action was used to define the portable software instance. This information identifies the download server and the location of the portable software instance files on that server.

The GIMGTPKG program uses this XML to download the portable software instance files. For more information, see [Content of SERVER data set in z/OS SMP/E Commands](#).

client-options-xml

Information about the local z/OS server used to download the files from a download server by the Add Portable Software Instance from Download Server action. The value is either the XML used by the GIMGTPKG program to download the portable software instance files, or it is the name of a data set or UNIX file that contains the XML. For more information, see [Content of CLIENT data set in z/OS SMP/E Commands](#).

jcl-data-set-name

Information from when the Add Portable Software Instance from Download Server action was used to define the portable software instance. This value lists the name of the data set that contains the generated JCL that was used to download the portable software instance files from a remote download server.

last-modified

Date and time, in ISO 8601 format, that the portable software instance was last modified.

modified-user-ID

User ID of the user who last modified the portable software instance.

date-created

Date and time, in ISO 8601 format, that the portable software instance was created.

created-user-ID

User ID of the user who created the portable software instance.

date-locked

Date and time, in ISO 8601 format, that the portable software instance was locked. This attribute is null if the portable software instance is not currently locked.

locked-user-ID

User ID of the user who locked the portable software instance. This attribute is null if the portable software instance is not currently locked.

signatureverificationkeyring

Identifies the location of the certificate that is used to validate the digital signature of the portable software instance. This is the name for a security manager keyring or "javatruststore". The value "javatruststore" indicates that the certificates in the default Java truststore were used to validate the digital signature. A Java truststore is a Java keystore file that contains the collection of the trusted Certificate Authority (CA) certificates. The default Java truststore is located relative to the Java home directory.

packagesignerdetails

Container for information that pertains to the certificate that is used to sign the portable software instance.

subjectname

The subject name of the certificate that is used to sign the portable software instance.

serialnumber

The serial number of the certificate that is used to sign the portable software instance.

fingerprint

The digital fingerprint of the certificate that is used to sign the portable software instance.

issuer

The subject name of the root certificate that issued the certificate that is used to sign the portable software instance.

products

Array that contains each product that is included in the portable software instance.

product-name

Name of the product. If any FMIDs are not related to a product and feature, those FMIDs are listed under a product that is named "No Product".

product-ID

Identifier of the product.

release-level

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

vendor-name

Name of the vendor that provides the product.

general-availability-date

Date on which this level of the product is available to all users.

end-of-service

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions.

product-URL

URL that links to additional information about the product. This information can include, for example, product lifecycle dates, product highlights, planning information, and technical descriptions.

file-version

Version of the most recently retrieved product information file that contains the corresponding product. The version represents the date on which the file was created or last updated.

features

Array that contains each feature that is included in the product.

feature-name

Name of the feature. If any FMIDs are not related to a product and feature, those FMIDs are listed under a feature that is named "No Feature".

nonsmpeproducts

List of products for the portable software instance that are not managed by SMP/E.

product-name

Name of the product. The value can contain up to 64 characters.

product-ID

Identifier for the product. The value can contain up to 64 characters.

release-level

Release level for the product. The value can contain up to 64 characters.

vendor-name

Name of the vendor that provides the product. The value can contain up to 64 characters.

product-URL

A URL that links to additional information about the product. The value can contain up to 256 characters.

feature-name

List of names of features for the product. The value can contain up to 64 characters.

general-availability-date

Date on which this level of the product is available to all users.

end-of-service-date

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. The following values are acceptable:

NotAnnounced

The end of service date is not yet announced for the product.

null

The end of service date is unknown for the product.

yyyy-mm-ddThh:mm:ssZ

The known end of service date, in ISO 8601 format.

Example

In the following example, the GET method is used to retrieve the properties of portable software instance Hooli on system PEV174.

```
GET /zosmf/swmgmt/pswi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com
```

Figure 231. Sample request to retrieve the properties of a portable software instance

A sample response is shown in [Figure 232 on page 525](#).

```

HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close

{
  "name": "DB2V9", "system": "PEV174", "description": null, "directory":
  "/u/fred/DB2V9", "categories": null, "lastmodified": "2014-08-20T19:23:25+00:00",
  "modifiedby": "FRED", "created": "2014-08-20T19:23:25+00:00", "createdby": "BARNEY",
  "locked": null, "lockedby": null, "products": [{"prodname": "DB2 for z/OS",
  "productid": "5635-DB2", "release": "09.01.00", "vendor": "IBM",
  "generalavailability": "20006-06-09T19:23:25+00:00", "endofservice": "2014-06-
  27T19:23:25+00:00", "url": null, "productinfofileversion": "2014-01-01",
  "features": [{"feature": "DB2 Base"}]}, {"nonsmpeproducts": [{"prodname":
  "Acme Software", "productid": "5990-ACME", "release": "100", "vendor":
  "Acme Software Products", "generalavailability": "2020-01-01T00:00:00Z",
  "endofservice": "2021-06-01T00:00:00Z", "url": "http://acmesoftware.com",
  "features": [{"Runtime"}]}],
  }

```

Figure 232. Sample response from a request to retrieve the properties of a portable software instance

Add a new portable software instance

You can use this operation to add a portable software instance to z/OSMF.

HTTP method and URI path

```
POST /zosmf/swmgmt/pswi
```

where:

zosmf/swmgmt

Identifies the software management services.

pswi

Informs the service that the request is for a portable software instance object.

Standard headers

Use the following standard HTTP headers with this request:

Content-Type

Identifies the type of input content that is provided by the caller. The following value is acceptable:

- Content-Type: application/json (JSON)

The JSON content type is used for the JSON document that is included as input with this request.

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

Your request must include a JSON object that describes the portable software instance to add and the configuration information for the target software instance. An example of the request content is shown in Figure 233 on page 526.

```
{
  "name": "pswi-name",
  "system": "system-nickname",
  "description": "pswi-description",
  "directory": "UNIX-directory",
  "categories": ["pswi-category"],
  "signaturekeyring": "keyring-name",
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 233. Adding a portable software instance: request content

where:

pswi-name

Name of the portable software instance. The name is required and must be unique on the system. The name can contain up to 30 non-blank characters, including the following alphanumeric characters, mathematical symbols, punctuation marks, and special characters: 0-9 A-Z a-z ! " # \$ ' - / : < = > ? | @ \ ^ _

system-nickname

Nickname of the system that has access to the volumes and data sets where the portable software instance is located. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems that are defined to z/OSMF, you can use the z/OSMF topology services. For more information, see [“Topology services”](#) on page 687.

pswi-description

Description of the portable software instance. The description can contain a maximum of 256 characters.

UNIX-directory

UNIX directory to contain the files for the portable software instance. Ensure that the directory name is valid according to the following UNIX directory name syntax:

- Must be an absolute path name.
- Must start and end with a forward slash.
- Can contain up to 1023 characters.

pswi-category

Comma-separated list of the categories to which the portable software instance is assigned. Each category name can contain up to 30 non-blank characters, including the following alphanumeric characters, mathematical symbols, punctuation marks, and special characters: 0-9 A-Z a-z ! " # \$ ' - / : < = > ? | @ \ ^ _

signaturekeyring

Identifies the location of the Certificate Authority (CA) certificate that is required to validate the digital signature of the portable software instance. The name for a security manager keyring or the keyword **javatruststore** may be specified.

keyring-name

Identifies the name for a security manager keyring. The specified name may be for a real or virtual keyring. Keyring names are from 1 to 237 characters in length and may include only characters X'40' through X'FE' excluding the forward slash (/), and the reserved XML characters, less than (<), greater than (>), double quotation mark ("), and ampersand (&).

If the keyring is associated with a userid other than the one that is used to perform the action, then the keyring name must be prefixed with the associated userid. Userids are from 1 to 8 alphanumeric characters in length, and can consist entirely of numerics and need not begin with any particular character. The userid is separated from the keyring value by a forward slash (userid/keyring).

To indicate that all of the Certificate Authority (CA) certificates that are defined in the security manager may be used to validate the digital signature, use the CERTAUTH virtual keyring by specifying the userid/keyring value “*AUTH*/”.

Note:

The userid that performs the action must be properly authorized to the FACILITY class resource IRR.DIGTCERT.LISTRING or to the RDATA LIB class resource keyring-owner.keyring-name.LST for z/OSMF to use the specified keyring. READ access is required for a userid to use its own keyring or the CERTAUTH virtual keyring. UPDATE access is required to use a keyring from another user. If the RDATA LIB class is used, for any real keyring, READ access to keyring-owner.keyring-name.LST is required. For the CERTAUTH virtual keyring, READ access to CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST is required.

javatruststore

Indicates that all of the certificates in the default Java truststore may be used to validate the digital signature. A Java truststore is a Java keystore file that contains the collection of the trusted Certificate Authority (CA) certificates. The default Java truststore is located relative to the Java home directory.

zosmf-user-ID

User ID for authenticating with a remote z/OSMF instance.

zosmf-password

Password for authenticating with a remote z/OSMF instance.

proxy-user-ID

User ID for authenticating with an HTTP proxy.

proxy-password

Password for authenticating with an HTTP proxy.

The request content is required, but some properties are optional. For example, consider whether the UNIX directory is not in the same sysplex as the primary z/OSMF instance. You might need to authenticate with the secondary z/OSMF instance that is running in the sysplex where the source software instance is located. In addition, consider whether the primary z/OSMF instance must access an HTTP proxy server to connect with the secondary z/OSMF instance. You might also need to authenticate with that HTTP proxy server. Therefore, you might need to specify the remote z/OSMF user ID and password and the proxy user ID and password.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

Certain authorizations are required to submit a request through the software management services to add a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform an add operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task and CONTROL access to the SAF resources that correspond to the portable software instance to add. If the specified categories are implicitly added during this portable software instance add operation, the user ID must also have CONTROL access to the SAF resources that correspond to the specified categories. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

Expected response

Validating the portable software instance files is an asynchronous operation. On completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted. The response also includes the following JSON object:

```
{
  "statusurl": "url"
}
```

where:

statusurl

Indicates the URL that provides status for the portable software instance add request. The client issues GET requests to the supplied URL to monitor the status of the operation and to obtain the result set.

Subsequent GET requests to the status monitor URL obtain the status of the operation. The following values are possible:

200 OK

The operation is either not yet complete or is complete with no errors. A JSON document is included that contains status information for the operation. It indicates either *running* or *complete*.

4nn or 5nn

An error occurred. For more information, see [“Error handling” on page 460](#). This document contains a reason code and a list of one or more message strings that describe the errors that are detected during request processing.

404 Not found

The request expired. When the operation completes, the z/OSMF server retains the result set for a finite length of time. After that time passes, the result set expires and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status"
  "packagesignerdetails": {
    "subjectname": "subject-name",
    "serialnumber": "serial-number",
    "fingerprint": "finger-print",
    "issuer": "issuer-subject-name"
  }
}
```

where:

status

The status of the Add portable software instance request. The status is either *running* or *complete*.

packagesignerdetails

Container for information that pertains to the certificate that is used to sign the portable software instance.

subjectname

The subject name of the certificate that is used to sign the portable software instance.

serialnumber

The serial number of the certificate that is used to sign the portable software instance.

fingerprint

The digital fingerprint of the certificate that is used to sign the portable software instance.

issuer

The subject name of the root certificate that issued the certificate that is used to sign the portable software instance.

Example

In the following example, the POST method is used to add a portable software instance to the z/OSMF instance that has a hostname of pev174.yourco.com.

```
POST /zosmf/swmgmt/pswi HTTP/1.1
Host: pev174.yourco.com
Content-Type: application/json
{
  "name": "Hooli",
  "system": "PEV174",
  "description": "Fictitious software product from Hooli",
  "directory": "/u/ibmusr/hoolipswi/"
}
```

Figure 234. Sample request to add a portable software instance

An example response is as follows:

```
HTTP/1.1 202 Accepted
Date: Fri, 5 June 2020 18:53:04 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl": "https://\pev174.yourco.com\zosmf\swmgmt\statusmonitor\addpswi\4837290198343"}
```

Figure 235. Response to sample request to add a portable software instance

Figure 235 on page 529 indicates that the request to validate and create a portable software instance entry was accepted, and the status monitor URL is provided.

A subsequent GET request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/addpswi/4837290198343 HTTP/1.1
Host: pev174.yourco.com
```

Figure 236. Sample GET request to the status monitor URL

An example response is as follows:

```
HTTP/1.1 200 OK
Date: Fri, 5 June 2020 18:54:04 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status": "running"}
```

Figure 237. Response to sample GET request to the status monitor URL

Figure 237 on page 529 indicates that the request to validate and create a portable software instance entry is still running.

A final request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/addpswi/4837290198343 HTTP/1.1
Host: pev174.yourco.com
```

Figure 238. Sample final request to the status monitor URL

An example response is as follows:

```
HTTP/1.1 200 OK
Date: Fri, 5 June 2020 18:55:04 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status": "complete"}
```

Figure 239. Response to sample final request to the status monitor URL

Figure 239 on page 530 indicates that the request to validate and create a portable software instance completed with no errors.

Delete a portable software instance

You can use this operation to remove a portable software instance definition from z/OSMF. The delete operation removes only the definition of the portable software instance from z/OSMF. The UNIX directory that contains the portable software instance files is not affected.

HTTP method and URI path

```
DELETE /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
```

where:

zosmf/swmgmt

Identifies the software management services.

pswi

Informs the service that the request is for a portable software instance object.

<system-nickname>

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located. In combination with *<pswi-name>*, it further qualifies the request and indicates the specific portable software instance to delete.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services”](#) on page 687.

<pswi-name>

Name of the portable software instance. In combination with *<system-nickname>*, it further qualifies the request and indicates the specific portable software instance to delete.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

Certain authorizations are required to submit a request through the software management services to delete a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform a remove operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task and CONTROL access to the SAF resources that correspond to the portable software instance to delete. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more information about errors, see [“Error handling” on page 460](#).

Example

In the following example, the DELETE method is used to delete portable software instance Hooli from system PEV174.

```
DELETE /zosmf/swmgmt/pswi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com
```

Figure 240. Sample request to delete a portable software instance

Figure 241 on page 531 provides a sample response that indicates that the delete operation was successful.

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
```

Figure 241. Sample response for a delete portable software instance request

Missing Critical Updates

The Missing Critical Updates service helps you determine if your software instances are missing software updates to resolve PE PTFs, HIPER fixes, or other exception SYSMODs identified by ERROR HOLDDATA, and helps you identify the SYSMODs that resolve those exceptions.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/missingcriticalupdates
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/missingcriticalupdates
```

Note: Before you use this service, use SMP/E to receive the latest HOLDDATA into the global zone associated with the software instance. Otherwise, the results might be incomplete, incorrect, or outdated. For more information, see [How to process HOLDDATA in z/OS SMP/E User's Guide](#).

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of **<system-nickname/swi-name>** to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).
- **/missingcriticalupdates** indicates that the Missing Critical Updates action is to be performed and the results returned.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document, if any, included as input with this request.

Custom headers

None.

Request content

Your request must include a JSON object that identifies properties that are required to perform the operation, for example:

```
{
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

Where:

zosmfuid

Userid for authenticating with a remote z/OSMF instance. This is an optional property.

zosmfpw

Password for authenticating with a remote z/OSMF instance. This is an optional property.

proxyuid

Userid for authenticating with an HTTP proxy. This is an optional property.

proxypw

Password for authenticating with an HTTP proxy. This is an optional property.

The request content is required only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you may also be required to authenticate with that HTTP proxy server. Therefore, you might need to specify the remote z/OSMF userid, password, and proxy userid and password.

Required authorizations

To submit a modify software instance request through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation that uses the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task and the SAF resources for the software instance that is being processed.

For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Usage considerations

For more information about usage considerations, see [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

The Missing Critical Updates action for a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document that contains a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example:

```
{
  "statusurl": "url"
}
```

statusurl

Indicates the URL that provides status for the missing critical updates request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document that contains status information for the operation.
- If the operation is complete, then an HTTP response code of 200 OK is returned, along with a JSON document that contains status information and the desired result set.
- If the request expires, then an HTTP response code of 404 Not found is returned. That is, when the operation has completed, the z/OSMF server holds the result set for a finite length of time. After that time has passed, the result set is said to expire, and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```

{
  "status": "status",
  "lastholdrecvdate": "date"
  "missingcriticalupdates": [{
    "name": "sysmod-id",
    "holdclass": "hold-class",
    "holdsymptom": "hold-symptom",
    "heldsysmod": "sysmod-id",
    "fmid": "fmid",
    "fmiddesc": "fmid-description",
    "resolvers": [{
      "name": "sysmod-id",
      "received": true | false
    }],
    "tgtzones": ["zone-name"]
  }]
}

```

status

The status of the report request. The status can be either "running" or "complete".

lastholdrecvdate

Date and time in ISO 8601 format when HOLDDATA was last received into the software instance's global zone.

missingcriticalupdates

List of critical software updates that are not yet installed in the software instance target zones.

name

SYSMOD ID for the missing update (APAR fix).

holdclass

Hold class that is specified on the CLASS operand of the ++HOLD statement.

holdsymptom

Short description of the problem fixed by the missing APAR.

heldsysmod

SYSMOD ID for the SYSMOD with the unresolved error HOLD and is fixed by the missing APAR.

fmid

FMID for the held SYSMOD.

fmiddesc

Description of the FMID.

resolvers

The list of SYSMODs that resolve the missing APAR. This list includes SYSMODs that supersede the missing APAR and resolving SYSMODs that are identified on the ++HOLD statement.

If this field is null, either no resolving SYSMODs are received in the global zone, or no resolving SYSMODs are identified for the missing APAR.

name

SYSMOD ID for the resolving SYSMOD.

received

Indicates whether the resolving SYSMOD is received in the software instance's global zone.

tgtzones

List of target zone names where the missing APAR is not installed and a resolving SYSMOD must be installed to resolve the critical problem.

For more information about error responses, see [“Error handling” on page 460](#).

Example

The request that is shown in [Figure 242 on page 535](#) performs the missing critical updates report for the software instance that is named `zos24` on `SYS123`.

```
POST /zosmf/swmgmt/swi/SYS123/zos24/missingcriticalfixes HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
```

Figure 242. Sample POST request for Missing Critical Updates

A sample response is shown in [Figure 243 on page 535](#):

```
HTTP/1.1 202 Accepted
Date: Tues, 1 February 2022 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/missingcriticalupdates/4837290198343"}
```

Figure 243. Sample POST response for Missing Critical Updates

The response in [Figure 243 on page 535](#) indicates that the request is accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is shown in [Figure 244 on page 535](#):

```
GET /zosmf/swmgmt/statusmonitor/missingcriticalupdates/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

Figure 244. Sample GET request for Missing Critical Updates

A sample response is shown in [Figure 245 on page 535](#):

```
HTTP/1.1 200 OK
Date: Tues, 1 February 2022 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{
  "status": "complete",
  "lastholdrecvdate": "2022-02-11T19:15:34Z",
  "missingcriticalupdates": [
    {
      "name": "CA61368",
      "holdclass": "HIPER",
      "holdsymptom": "FUL",
      "heldsysmod": "HBB77C0",
      "fmid": "HBB77C0",
      "fmiddesc": "BCP Base",
      "resolvers": [
        {
          "name": "UJ07031",
          "received": true
        }
      ],
      "tgtzones": ["TGT24"]
    },
    {
      "name": "CA62658",
      "holdclass": "PE",
      "holdsymptom": null,
      "heldsysmod": "UJ00132",
      "fmid": "HBB77C0",
      "fmiddesc": "BCP Base",
      "resolvers": null,
      "tgtzones": ["TGT24"]
    }
  ]
}
```

Figure 245. Sample GET response for Missing Critical Updates

Missing FIXCAT Updates

The Missing FIXCAT Updates service helps you identify missing updates for fix categories that might be applicable to the software instance, and it identifies the SYSMODS that can resolve the missing updates.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/missingfixcatupdates
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/missingfixcatupdates
```

Note: Before you use this service, use SMP/E to receive the latest HOLDDATA into the global zone associated with the software instance. Otherwise, the results might be incomplete, incorrect, or outdated. For more information, see [How to process HOLDDATA in z/OS SMP/E User's Guide](#).

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 687](#).

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF” on page 461](#).
- **/missingfixcatupdates** indicates that the Missing FIXCAT Updates action is to be performed and the results returned.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document, if any, included as input with this request.

Custom headers

None.

Request content

Your request must include a JSON object that identifies properties that are required to perform the operation, for example:

```
{
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

Where:

zosmfuid

Userid for authenticating with a remote z/OSMF instance. This is an optional property.

zosmfpw

Password for authenticating with a remote z/OSMF instance. This is an optional property.

proxyuid

Userid for authenticating with an HTTP proxy. This is an optional property.

proxypw

Password for authenticating with an HTTP proxy. This is an optional property.

The request content is required only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you may also be required to authenticate with that HTTP proxy server. Therefore, you might need to specify the remote z/OSMF userid, password, and proxy userid and password.

Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when you perform an analogous operation that uses the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task and the SAF resources for the software instance that is being processed.

For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Usage considerations

For more information about usage considerations, see [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

The Missing FIXCAT Updates action for a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document that contains a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example,:

```
{
  "statusurl": "url"
}
```

statusurl

Indicates the URL that provides status for the missing critical updates request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document that contains status information for the operation.
- If the operation is complete, then an HTTP response code of 200 OK is returned, along with a JSON document that contains status information and the desired result set.
- If the request expires, then an HTTP response code of 404 Not found is returned. That is, when the operation has completed, the z/OSMF server holds the result set for a finite length of time. After that time has passed, the result set is said to expire, and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",
  "lastholdrecvdate": "date"
  "missingfixcatupdates": [{
    "name": "sysmod-id",
    "fixcats": ["fix-category"],
    "fmid": "fmid",
    "fmiddesc": "fmid-description",
    "resolvers": [{
      "name": "sysmod-id",
      "received": true | false
    }],
    "tgtzones": ["zone-name"]
  }]
}
```

status

The status of the report request. The status can be either "running" or "complete".

lastholdrecvdate

Date and time in ISO 8601 format when HOLDDATA was last received into the software instance's global zone.

missingfixcatupdates

List of software updates that are associated with one or more fix categories, which are not yet installed in the software instance target zones.

name

SYSMOD ID for the missing APAR.

fixcats

List of fix categories associated with the missing APAR.

fmid

FMID for the missing APAR.

fmiddesc

Description of the FMID.

resolvers

The list of SYSMODs that resolve the missing APAR. This list includes SYSMODs that supersede the missing APAR and resolving SYSMODs that are identified on the ++HOLD statement.

If this field is null, either no resolving SYSMODs are received in the or no resolving SYSMODs are identified for missing APAR.

name

SYSMOD ID for the resolving in the global zone, g SYSMOD.

received

Indicates whether the resolving SYSMOD is received in the software instance's global zone.

tgtzones

List of target zone names where the missing APAR is not installed and a resolving SYSMOD must be installed to resolve the critical problem.

For more information about error responses, see [“Error handling” on page 460](#).

Example

The request that is shown in [Figure 246 on page 539](#) performs the missing FIXCAT updates for the software instance that is named zos24 on SYS123.

```
POST /zosmf/swmgmt/swi/SYS123/zos24/missingfixcatupdates HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
```

Figure 246. Sample POST request for Missing FIXCAT Updates

A sample response is shown in [Figure 247 on page 539](#):

```
HTTP/1.1 202 Accepted
Date: Tues, 1 February 2022 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/missingfixcatupdates/4837290198343"}
```

Figure 247. Sample POST response for Mission FIXCAT Updates

The response in [Figure 247 on page 539](#) indicates that the request is accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is shown in [Figure 248 on page 539](#):

```
GET /zosmf/swmgmt/statusmonitor/missingfixcatupdates/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

Figure 248. Sample GET request for Missing FIXCAT Updates

A sample response is shown in [Figure 249 on page 540](#) :

```

HTTP/1.1 200 OK
Date: Tues, 1 February 2022 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{
  "status": "complete",
  "lastholdrecvddate": "2022-02-11T19:15:34Z"
  "missingfixcatupdates": [
    {
      "name": "CA61368",
      "fixcats": ["IBM.ProductInstall-RequiredService"],
      "fmid": "HBB77C0",
      "fmiddesc": "BCP Base",
      "resolvers": [
        {
          "name": "UJ07031",
          "received": true
        }
      ],
      "tgtzones": ["TGT24"]
    },
    {
      "name": "CA62202",
      "fixcats": ["IBM.ProductInstall-RequiredService"],
      "fmid": "HBB77C0",
      "fmiddesc": "BCP Base",
      "resolvers": [
        {
          "name": "UJ07520",
          "received": true
        }
      ],
      "tgtzones": ["TGT24"]
    }
  ]
}

```

Figure 249. Sample GET response for Missing FIXCAT Updates

Software Update Search

The Software Update Search service allows you to search a software instance for one or more software updates. This is helpful when you need to complete a task that requires you to determine if a software instance is installed or needs to be installed.

HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/softwareupdatesearch
```

```
POST /zosmf/swmgmt/swi/<swi-uuid>/softwareupdatesearch
```

Where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services”](#) on page 687.

In addition, the software instance UUID can be used instead of *<system-nickname/swi-name>* to identify a software instance.

- **<swi-uuid>** further qualifies the request and indicates the specific software instance to be analyzed. A software instance can be uniquely identified by using its UUID. A UUID is assigned to every software instance and can be obtained by using the REST API [“List the software instances defined to z/OSMF”](#) on page 461.

- **/softwareupdatesearch** indicates that the SYSMOD Search service is to be performed and the results returned.

Standard headers

Use the following standard HTTP header with this request:

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document, if any, included as input with this request.

Custom headers

None.

Request content

Your request must include a JSON object that identifies properties that are required to perform the operation, for example:

```
{
  "updates": ["update-id"],
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

Where:

updates

The list of software update (SYSMOD) IDs to be searched in the subject software instance.

zosmfuid

Userid for authenticating with a remote z/OSMF instance. This is an optional property.

zosmfpw

Password for authenticating with a remote z/OSMF instance. This is an optional property.

proxyuid

Userid for authenticating with an HTTP proxy. This is an optional property.

proxypw

Password for authenticating with an HTTP proxy. This is an optional property.

The request content is required, but some properties are optional. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. Therefore, you may need to specify the remote z/OSMF userid, password, and proxy userid and password.

Required authorizations

To submit a modify software instance request through the software management services, the user ID initiating the request requires the same authorizations as when you perform an analogous operation that uses the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task and the SAF resources for the software instance that is being processed.

For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

Usage considerations

For more information about usage considerations, see [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

The software updates search action for a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document that contains a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example:

```
{
  "statusurl": "url"
}
```

statusurl

Indicates the URL that provides status for the missing critical updates request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document that contains status information for the operation.
- If the operation is complete, then an HTTP response code of 200 OK is returned, along with a JSON document that contains status information and the desired result set.
- If the request expires, then an HTTP response code of 404 Not found is returned. That is, when the operation has completed, the z/OSMF server holds the result set for a finite length of time. After that time has passed, the result set is said to expire, and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",
  "updates": [
    {
      "name": "sysmod-id",
      "type": "sysmod-type",
      "fmid": "fmid",
      "zones": [
        {
          "zone": "zone-name",
          "status": "sysmod-status",
          "installed": "date-time"
        }
      ]
    }
  ]
}
```

status

The status of the report request. The status can be either "running" or "complete".

updates

List of software updates (SYSMODs) that were searched.

name

Software update (SYSMOD) ID that was searched.

type

Type of SYSMOD. The SYSMOD can be one of the following types: APAR, FUNCTION, PTF, or USERMOD.

fmid

FMID for the subject SYSMOD.

zones

The list of zones in the software instance.

zone

Name of a global, target, or dlib zone.

status

Status of the SYSMOD in this zone. May be one of the following values:

- Accepted. The SYSMOD is installed in the distribution zone.
- Applied. The SYSMOD is installed in the target zone.
- Deleted. The SYSMOD was deleted when another SYSMOD was installed.
- Error. The SYSMOD has only been partially installed. Errors occurred during APPLY or ACCEPT processing.
- Not Found. The SYSMOD was not found in the corresponding zone.
- Received. An entry for the SYSMOD exists in the global zone.
- Superseded by <sysmod-id>. A SYSMOD that supersedes the subject SYSMOD is installed in the target or distribution zone. The ID of the superseding SYSMOD is provided.

installed

Date and time the SYSMOD was received, applied, or accepted into the global zone, target zone, or distribution zone, respectively.

For more information about error responses, see [“Error handling” on page 460](#).

Example

The request that is shown in [Figure 250 on page 543](#) performs the software update search action for the software instance that is named zos24 on SYS123.

```
POST /zosmf/swmgmt/swi/SYS123/zos24/softwareupdatesearch HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "updates": ["HBB77C0", "UA12345", "UJ00132", "CA57408"]
}
```

Figure 250. Sample POST request for Software Update Search

A sample response is shown in [Figure 251 on page 543](#):

```
HTTP/1.1 202 Accepted
Date: Tues, 1 February 2022 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl": "https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/softwareupdatesearch/4837290198343"}
```

Figure 251. Sample POST response for Software Update Search

The response in [Figure 251 on page 543](#) indicates that the request is accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is shown in [Figure 252 on page 544](#):

```
GET /zosmf/swmgmt/statusmonitor/softwareupdatesearch/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

Figure 252. Sample GET request for Software Update Search

A sample response is shown in [Figure 253 on page 544](#):

```
HTTP/1.1 200 OK
Date: Tues, 1 February 2022 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{
  "status": "complete",
  "updates": [
    {
      "name": "HBB77C0",
      "type": "FUNCTION",
      "fmid": "HBB77C0",
      "zones": [
        {
          "zone": "DLB24", "status": "Accepted", "installed": "2022-02-11T19:15:34Z"
        },
        {
          "zone": "GLOBAL", "status": "Not Found", "installed": null
        },
        {
          "zone": "TGT24", "status": "Applied", "installed": "2022-02-11T19:15:34Z"
        }
      ]
    },
    {
      "name": "UA12345",
      "type": null,
      "fmid": null,
      "zones": [
        {
          "zone": "DLB24", "status": "Not Found", "installed": null
        },
        {
          "zone": "GLOBAL", "status": "Not Found", "installed": null
        },
        {
          "zone": "TGT24", "status": "Not Found", "installed": null
        }
      ]
    },
    {
      "name": "UJ00132",
      "type": "PTF",
      "fmid": "HBB77C0",
      "zones": [
        {
          "zone": "DLB24", "status": "Not Found", "installed": null
        },
        {
          "zone": "GLOBAL", "status": "Received", "installed": "2022-02-11T18:34:56Z"
        },
        {
          "zone": "TGT24", "status": "Applied", "installed": "2022-02-11T19:15:34Z"
        }
      ]
    },
    {
      "name": "CA57408",
      "type": null,
      "fmid": null,
      "zones": [
        {
          "zone": "DLB24", "status": "Not Found", "installed": null
        },
        {
          "zone": "GLOBAL", "status": "Not Found", "installed": null
        },
        {
          "zone": "TGT24", "status": "Superseded by UJ00132", "installed": null
        }
      ]
    }
  ]
}
```

Figure 253. Sample GET response for Software Update Search

Retrieve the z/OS system UUID

You can use this operation to retrieve the UUID for the software instance that represents the installed software for the specified z/OSMF host system.

HTTP method and URI path

```
POST /zosmf/swmgmt/system/uuid/<system-nickname>
```

Where:

- **zosmf/swmgmt** identifies the software management services.
- **system/uuid** informs the service that the request is for obtaining a z/OS system's software instance UUID.

- **<system-nickname>** further qualifies the request and indicates the z/OSMF host system for which the software instance UUID is retrieved. This value is optional. If the system nickname is not specified, the UUID for the system that processes the REST API request is returned.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 687](#).

When you issue this request, z/OSMF reads module IZUSIUI from LPA to obtain the UUID value recorded in the module. It then returns the UUID value to the caller.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

Identifies the type of input content that is provided by the caller. Use the JSON content type ("Content-Type: application/json") if a JSON document is included as input with this request.

Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

Custom headers

None.

Request content

If the identified z/OSMF host system is not the primary z/OSMF instance, you might be required to authenticate with the remote z/OSMF instance. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the remote z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. To do so, include the following JSON object in your request:

```
{
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Where:

zosmf-user-ID

User ID for authenticating with the remote z/OSMF instance.

zosmf-password

Password for authenticating with the remote z/OSMF instance.

proxy-user-ID

User ID for authenticating with the HTTP proxy server.

proxy-password

Password for authenticating with the HTTP proxy server.

Include the JSON object in the request only if you are required to authenticate with a remote z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

To retrieve the UUID for a z/OSMF host system, the user ID must have READ access to the Software Management task.

Expected response

On completion, the service returns an HTTP response, which includes a status code, which indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error has occurred. For more details, see [“Error handling”](#) on page 460.

If the request is successful, the response also includes the following JSON object:

```
{
  "uuid": "uuid-value"
  "modid": "modid-value"
}
```

Where:

uuid-value

UUID for the software instance that corresponds with the installed software for the specified z/OSMF host system. The value can be null if no UUID is defined for the requested z/OSMF host system.

modid-value

Includes the UUID module name and assembly date. The value can be null if no UUID is defined for the requested z/OSMF host system.

Example

In the following example, the POST method is used to retrieve the UUID for system PEV174.

```
POST /zosmf/swmgmt/uuid/PEV174 HTTP/1.1
Host: zosmf1.yourco.com
```

A sample response is shown in [Figure 254 on page 546](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2023 05:39:28 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close
{
  "uuid": "bbc9e8d6-bd61-4f11-af48-ff500fffc178"
  "modid": "IZUSIUI 2023.032 "
}
```

Figure 254. Sample response from a request to retrieve the z/OS system UUID

Storage management services

The storage management services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. The storage management services provide a programming interface for system storage elements.

Table 335 on page 547 lists the operations that the storage management services provide.

Table 335. Operations provided through the storage management services.	
Operation	HTTP method and URI path ¹
“Get a list of storage groups” on page 549	GET /zosmf/storage/rest/<version>/storagegroups
“Get a storage group definition” on page 560	GET /zosmf/storage/rest/<version>/cds/<scds-name>
“Get a list of volumes” on page 571	GET /zosmf/storage/rest/<version>/volumes
“Get a volume definition” on page 576	GET /zosmf/storage/rest/<version>/volumes/<volume-ser>
“Get a list of data classes” on page 580	GET /zosmf/storage/rest/<version>/dataclasses
“Get a data class definition” on page 597	GET /zosmf/storage/rest/<version>/dataclasses/<dc-name>
“Get a list of storage classes” on page 614	GET /zosmf/storage/rest/<version>/storageclasses
“Get a storage class definition” on page 621	GET /zosmf/storage/rest/<version>/storageclasses/<sc-name>
“Define a volume list to a storage group” on page 628	POST /zosmf/storage/rest/<version>/storagegroups/<stg-name>/volumes
“Validate an SCDS” on page 634	PUT /zosmf/storage/rest/<version>/cds/<scds-name>
“Activate an SCDS” on page 636	POST /zosmf/storage/rest/<version>/cds/<scds-name>
“Get an SCDS activation result” on page 638	GET /zosmf/storage/rest/<version>/cds/<scds-name>

Notes:

1. The valid value for <version> is v1.

Using the Swagger interface

You can use the Swagger interface to display information about the storage management services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Common HTTP Request Headers

Accept-Encoding

This request header indicates which forms of content encoding (compression types) are acceptable to the client program. The actual encoding that is used by the server is indicated in the Content-Encoding response header. The supported encoding method is gzip, which is specified as follows:

```
Accept-Encoding: gzip
```

The response content is compressed by this method, if the content exceeds 4096 bytes.

Common HTTP Response Headers

Content-Encoding

This response header indicates which encoding method was used to decompress the media-type that was specified in the Accept-Encoding header. The supported encoding method is gzip, which is specified as follows:

```
Content-Encoding: gzip
```

The response content is compressed by this method, if the content exceeds 4096 bytes.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200 OK

The request was processed successfully.

HTTP 200 Created

The request was created successfully.

HTTP 204 (No content)

The request was completed successfully.

HTTP 400 Bad request

The request cannot be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 403 Forbidden

The server received the request but rejected it.

HTTP 404 Not found

The requested resource does not exist.

HTTP 405 Method not allowed

The requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

HTTP 408 Request timed out

The client did not produce a request within the allowed time. The request can be submitted again.

HTTP 500 Server error

The server encountered an error when it processed the request. For a more specific indication of the error, check the response for a reason code.

HTTP 501 Not implemented

The request specifies an HTTP method that is not recognized by the server.

HTTP 503 Service unavailable

The request cannot be completed by the server because of a temporary condition. If the response contains a Retry-After header, check the suggested wait time. Otherwise, the requester can treat the response as an HTTP 500 response.

Some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error reporting categories” on page 639](#).

Error logging

Errors from the storage management services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Get a list of storage groups

This operation retrieves a list of storage group records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all storage groups that match the query parameters.

HTTP method and URI path

GET /zosmf/storage/rest/<version>/storagegroups

Query parameters

You can include query parameters on this request to filter storage groups and determine the level of detail in the returned content. For descriptions of the query parameters, see [“Query parameters” on page 549](#).

Table 336. Query parameters for retrieving a list of storage groups	
Query Parameter	Description
detail-data	Specifies whether the response contains detailed information. Valid values are Y and N. A Y value requests detailed information. A N value requests summary information. The default value is N. (N is used if Y is not specified.)

Table 336. Query parameters for retrieving a list of storage groups (continued)

Query Parameter	Description
filter	<p>Specifies a storage-group name.</p> <p>The value is not case-sensitive.</p> <p>The value must meet the following criteria:</p> <ul style="list-style-type: none"> • Contains 1-8 alphanumeric characters • Begins with an alphabetic or national character. National characters are \$, #, and @. • The asterisk (*) and percent (%) wildcards are allowed. <ul style="list-style-type: none"> – Asterisk matches any number of characters. – Percent matches one character. – Asterisk and percent wildcards can be used together. – You can use more than one asterisk and more than one percent wildcard.
type	<p>Specifies a storage group type. The value must be one of the following types:</p> <ul style="list-style-type: none"> • VIO • Pool • Dummy • Copy Target • Object • Object backup • Tape
volume-ser	<p>Specifies a volume serial number that is associated with a storage group.</p> <p>The value must be a complete serial number, 1 - 6 characters, no wildcard characters. Partial volume numbers are not matched. The value is not case-sensitive.</p> <p>If volume-ser is specified, the filter, type, offset, and limit values are ignored.</p>
offset	<p>Specifies the offset for the list of storage groups. The request retrieves storage groups starting from this offset value. For example, an offset of 50 means retrieve from the 51st position in the list and onward. The valid range of offset values is 0 - 2147483647.</p> <p>The offset cannot be larger than or equal to the number of storage groups.</p> <p>The offset cannot be used with the filter parameter in the same request.</p> <p>The default value for offset is 0.</p>

Table 336. Query parameters for retrieving a list of storage groups (continued)

Query Parameter	Description
limit	<p>Specifies the maximum number of storage groups that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>You can use the limit parameter with the offset parameter. For example, if you specify offset = 1 and limit = 3, the request returns the second, third, and fourth storage groups from the result set.</p> <p>Observe the following usage notes:</p> <ul style="list-style-type: none"> • If you specify limit = 0, the request returns all of the queried storage groups from the offset value. • If the offset plus limit is a larger value than the number of queried storage groups, the request returns only the number of the queried storage groups, minus the offset. • If the offset plus limit is a smaller value than the number of queried storage groups, the request returns only the limit number of storage groups, starting from the offset. <p>The limit cannot be used with the filter parameter in the same request.</p> <p>The default value for limit is 0.</p>

Description

On successful completion, the request returns a list of storage group records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all storage groups that match the query parameters. For description of the query parameters that you can specify on the request, see [“Query parameters” on page 549](#).

For a description of the response content, see [“Response content: Summary record” on page 551](#) and [“Response content: Detailed record” on page 553](#).

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains an array of JSON storage group documents that match the specifications. If you do not request detailed information, each document contains only the information in [“Response content: Summary record” on page 551](#). If you request detailed information, each document contains the information in [“Response content: Detailed record” on page 553](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 337 on page 552](#).

Table 337. Response content for a successful retrieval of storage-group summary information

Field name	Description	Data type
description	Indicates the description of the storage group.	String
lastUser	Indicates the user ID of the user that made the last update to the storage group definition.	String
numberOfVolumes	Indicates the number of volume entries in the storage group's volume list. The number includes predefined volume serials, which are volume entries for which no physical volume exists.	Number
spaceAvailable	Indicates the total amount of available (free) capacity for all online volumes in the storage group, in gigabytes. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
spaceUsed	Indicates the total amount of used (allocated) capacity for all online volumes in the storage group, in gigabytes.	Number
status	<p>The status object is an array of four string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the storage group. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	Array
storageGroupName	<p>Indicates the storage group name.</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String

Table 337. Response content for a successful retrieval of storage-group summary information (continued)

Field name	Description	Data type
storageGroupType	Indicates the storage group type. The value is one of the following types: <ul style="list-style-type: none"> • VIO • Pool • Dummy • Copy Target • Object • Object backup • Tape 	String
totalSpace	Indicates the total capacity that is obtained from all online volumes in the storage group, in gigabytes. Only volumes that are connected to a valid system, for storage groups in the active configuration, are included. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
updateDate	Indicates the date that the storage group definition was last changed.	String
updateTime	Indicates the time that the storage group definition was last changed.	String

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 338 on page 553](#).

Table 338. Response content for a successful retrieval of storage-group detailed information

Field name	Description	Data Type
autoBackupSystem	Indicates the name of the system on which automatic backup of the volumes in the storage group takes place.	String
autoDump	Indicates whether the volumes in this storage group can be automatically dumped or automatically backed up by using DFSMSHsm or a comparable product.	Boolean
autoDumpSystem	Indicates the name of the system on which automatic dumping of the volumes in the storage group takes place.	String
autoMigrateSystem	Indicates the name of the system on which automatic migration and space management of the volumes in this storage group takes place. If the field is blank, automatic migration or space management can take place on any system.	String
autoMigration	Indicates whether data sets on volumes in this storage group can be moved to DASD or tape by the primary space management and interval migration functions of DFSMSHsm.	Boolean

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
breakPointValue	<p>Indicates the EAV breakpoint value, which is used in determining whether a data set can be allocated in a cylinder-managed space. Valid values are 0-65.</p> <p>The breakpoint value is expressed as a number of cylinders.</p> <p>When a disk space request is this size or larger, the system gives the cylinder-managed space preference for that extent. The preference applies to each request for primary or secondary space for data sets that are eligible for the cylinder-managed space. If not enough cylinder-managed space is available, then the system uses the track-managed space or uses both areas. When the size of a disk space request is less than the breakpoint value, the system gives the track-managed space preference. If enough space is not available there, the system uses the cylinder-managed space or uses both areas.</p>	Number
breakPointValueSpecified	Indicates whether the EAV breakpoint value is defined.	Boolean
driveStartThresholdBit	Indicates the maximum number of outstanding object write requests for an optical drive in this storage group. Valid values are 0-127.	Number
dumpClassesForAD	Indicates the names of the dump classes that are assigned to this storage group.	String
EAVTrackAllocationThresholdSpecified	Indicates whether the EAV track allocation threshold is defined.	Boolean
extendSgexname	Indicates the name of another pool storage group to which data sets from the primary storage group can be extended when the primary storage group has an insufficient amount of storage space.	String
extendSgexnameLength	Indicates the length of the storage group extend name. Valid values are 0-30.	Number
extendSGSpecified	Indicates whether data sets from the primary storage group can be extended when the primary storage group has an insufficient amount of storage space.	Boolean
extendSgtcname	Indicates the name of the copy-pool backup storage-group that contains volumes that are eligible for fast replication backup versions. The storage group can use the backup storage group only if the value of targetCopySGSpecified is true.	String
extendSgtcnameLength	Indicates the length of the target copy storage group name. Valid values are 0-30.	Number
guaranteedBackupFreq	<p>Indicates the maximum number of days between backups. The property is valid only for pool storage groups. Valid values are 1-9999.</p> <p>The property is ignored if the value of guaranteedBackupFreqNoLimit is "ON".</p>	Number

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
guaranteedBackupFreqNoLimit	Indicates whether the management class specifies how often data sets are backed up. Valid values are "ON" and "OFF". If the value is "OFF", the guaranteedBackupFreq value indicates the guaranteed backup frequency. If the value is "ON", the management class defines backup frequency.	String
guaranteedBackupFreqSpecified	Indicates whether the maximum number of days between backups is defined.	Boolean
highThreshold	Indicates the maximum space limit for the DASD volumes in a pool storage group. Valid values are 1-100.	Number
highTrackThreshold	Indicates the high threshold percent for the track-managed portion of the volume. Valid values are 1-100.	Number
HSMAutoBackup	Indicates whether the data sets on the volumes in the storage group are eligible for automatic backup.	Boolean
intervalMigration	Indicates whether interval migration is defined.	Boolean
lastUser	Indicates the user ID of the user that made the last update to the storage group definition.	String
libraryName	Indicates the optical library name.	String
libraryNameLength	Indicates the optical library name length.	Number
lowThreshold	Indicates the low threshold percent for the track-managed portion of the volume. The value applies to the primary storage group. Valid values are 0-99.	Number
lowTrackThreshold	Indicates the threshold percentage of space allocation that stops migration of data sets from volumes in this storage group during interval migration. The value applies to the track-managed space of an extended address volume. Valid values are 0-99.	Number
numberOfVolumes	Indicates the number of volume entries in the storage group's volume list. The number includes predefined volume serials, which are volume entries for which no physical volume exists.	Number
oamCycleEndtime	Indicates the last hour of a window of time in which object processing can be started for this storage group. Valid values are 0-23 and "None".	String
oamCycleStarttime	Indicates the first hour of a window of time in which object processing can be started for this storage group. Valid values are 0-23 and "None".	String

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
oamFlags	<p>Indicates the OAM flags for the storage group. Valid values are 0-127. The following properties are encoded:</p> <ul style="list-style-type: none"> • X'80': Whether OAM cycle start and end times are given. The times are defined in the oamCycleStartTime and oamCycleEndTime fields. • X'40': "Volume full bit": Whether optical volumes are marked as full when they have insufficient free space. The required free space is defined in the volumeFullThresholdBit field. • X'20': "Drive start threshold bit": Whether a maximum limit is defined for the number of outstanding object write requests for an optical drive in this storage group. The maximum limit is defined in the driveStartThresholdBit field. • X'10': Whether a "mark volume full on first write failure" criteria is defined for the storage group. • X'08': Whether retention protection is enabled for an object. When retention protection is enabled, OAM prevents an object from being deleted before the object's expiration date. Additionally, OAM prevents the expiration date from being changed to an earlier date. The expiration date can be changed to a later date. • X'04': Whether deletion protection is enabled for an object. When deletion protection is enabled, OAM prevents an object from being deleted before the object's expiration date. Deletion protection differs from retention protection in that the deletion protection can be turned on and off by the installation, and deletion protection does not restrict any changes to the expiration date. • X'02': This bit is reserved. • X'01': This bit is reserved. 	Number
oamTableSI	Indicates the OAM table space identifier for this storage group in the form GROUPnn.	String
osmcSystemName	Indicates which OAM system in the sysplex completes OSMC processing for this object storage group.	String
overflowSG	If overflow designation is defined, this property indicates whether this pool storage group is designated as an overflow storage group. An overflow storage group handles periods of high demand for initial primary space allocations. Volumes in overflow storage groups are selected for primary space allocation only when all the volumes in non-overflow storage groups cannot satisfy the allocation amount without exceeding the high threshold for the storage group.	Boolean
OverflowSGSpecified	Indicates whether overflow designation is defined.	Boolean

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
primarySpaceManagement	<p>Indicates whether data sets are eligible for primary space management only. If the value is TRUE, then the following eligibility applies:</p> <ul style="list-style-type: none"> • Data sets are eligible for primary space management (daily migration) only. • Data sets are not eligible for interval migration (hourly migration) even if Setsys Intervalmigration is specified in DFSMSHsm. <p>For space management processing, DFSMSHsm processes storage groups in the order of their processing priority. Storage groups that are migration targets for class transition processing require free space for the transitions. Those storage groups must have a relatively high processing priority.</p>	Boolean
processingPriority	<p>Indicates the space management processing priority. Valid values are 1-100.</p> <p>For space management processing, DFSMSHsm processes storage groups in the order of their processing priority. Storage groups that are migration targets for class transition processing require free space for the transitions. Those storage groups must have a relatively high processing priority.</p>	Number
processingPrioritySpecified	Indicates whether processing priority is specified.	Boolean
SMADDataSpecified	The SMS IPCS verb exit (SMSDATA) is intended for the use of diagnosticians who are working with the IBM Support Center to resolve an SMS-related problem.	Boolean
spaceAvailable	Indicates the total amount of available (free) capacity for all online volumes in the storage group, in gigabytes. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
spaceUsed	Indicates the total amount of used (allocated) capacity for all online volumes in the storage group, in gigabytes.	Number

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
status	<p>The status object is an array of four string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the storage group. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	Array
storageGroupName	<p>Indicates the storage group name.</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String
storageGroupType	<p>Indicates the storage group type. The value is one of the following types:</p> <ul style="list-style-type: none"> • VIO • Pool • Dummy • Copy Target • Object • Object backup • Tape 	String

Table 338. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
targetCopySGSpecified	Indicates whether the storage group can use a copy-pool backup storage-group for fast-replication backup versions. When the value is true, the storage group that is defined in extendSGtcname is valid.	Boolean
thresholdsSpecified	Indicates whether space thresholds are specified.	Boolean
tmsTotalSpaceAlert	Indicates the point at which storage use as a percentage of total storage exceeds the acceptable level. The value applies among the storage group's TAPE type storage.	Number
totalSpace	Indicates the total capacity that is obtained from all online volumes in the storage group, in gigabytes. Only volumes that are connected to a valid system, for storage groups in the active configuration, are included. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
totalSpaceAlert	Indicates the point at which storage use for the storage group, as a percentage of total storage, exceeds the acceptable level. The value applies among the storage group's TAPE type storage. Valid values are 0-99.	Number
updateDate	Indicates the date that the storage group definition was last changed.	String
updateTime	Indicates the time that the storage group definition was last changed.	String
vioMaxDss	Indicates the largest size of a virtual input/output (VIO) data set that you can create for this storage group, in kilobytes. You cannot allocate data sets that exceed this size in this Storage Group. This value applies to VIO storage groups only.	Number
vioUnitType	Indicates the type of physical device that is simulated by the storage group. At least one unit of the device type must be physically connected to each system that has access to the storage group. This value applies to VIO storage groups only.	String
volumeFullThresholdBit	Indicates the number of free sectors that are required for an optical volume in the storage group. When the number of free sectors in a volume becomes less than the threshold, the Object Access Method marks the optical volume as full. This value is valid only for the OBJECT and OBJECT BACKUP storage group types. Valid values are 0-127.	Number

Example HTTP interaction

1. The example in [Figure 255 on page 559](#) shows a request to get a list of storage groups of type pool and filter DOMUS.

```
GET zosmf/storage/rest/v1/storagegroups?filter=DOMUS&type=pool
```

Figure 255. Sample request to get a list of storage groups

The following code is the response body for the request.

```
[
  {
    "storageGroupName": "DOMUS",
    "spaceAvailable": 0,
    "lastUser": "KNMORT",
    "updateDate": "2003/08/07",
    "numberOfVolumes": 15,
    "spaceUsed": 0,
    "description": "STORAGE GROUP FOR DOMINO WORKLOAD",
    "updateTime": "11:19",
    "totalSpace": 0,
    "storageGroupType": "POOL",
    "status": [
      {
        "sysType": "1",
        "confirmedSmsStatus": "DISABLED/ALL",
        "sysName": "S1I",
        "requestedSystemStatus": "ENABLED"
      },
      ...
      {
        "sysType": "2",
        "confirmedSmsStatus": "DISABLED/ALL",
        "sysName": "XCFLOCAL",
        "requestedSystemStatus": "ENABLED"
      }
    ]
  }
]
```

Figure 256. Sample response body

Get a storage group definition

This operation retrieves a storage group definition. The response content contains properties of the storage group.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/cds/<scds-name>
```

Query parameters

You can include query parameters on this request to specify the storage group name and filter the level of detail in the returned content. See [Table 339 on page 560](#) for a description of the query parameters.

Table 339. Query parameters for retrieving a storage group definition

Query Parameter	Description	Required or Optional	Type
detail-data	Specifies whether the response contains detailed information. Valid values are Y and N. A Y value requests detailed information. A N value requests summary information. The default value is N. (N is used if Y is not specified.)	Optional	String

Table 339. Query parameters for retrieving a storage group definition (continued)

Query Parameter	Description	Required or Optional	Type
stg-name	Specifies a storage group name. The value must uniquely match only one storage group name. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none"> Contains 1 - 8 alphanumeric characters Begins with an alphabetic or national character. National characters are \$, #, and @. Wildcards are not allowed. 	Required	String

Description

On successful completion, the request returns the properties of a storage group. The response body contains summary information or detailed information, according to the query parameters.

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains a JSON storage group document. If you did not request detailed information, the document contains only the information in [“Response content: Summary record” on page 561](#). If you requested detailed information, the document contains the information in [“Response content: Detailed record” on page 563](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 340 on page 561](#).

Table 340. Response content for a successful retrieval of storage-group summary information

Field name	Description	Data type
description	Indicates the description of the storage group.	String
lastUser	Indicates the user ID of the user that made the last update to the storage group definition.	String
numberOfVolumes	Indicates the number of volume entries in the storage group's volume list. The number includes predefined volume serials, which are volume entries for which no physical volume exists.	Number
spaceAvailable	Indicates the total amount of available (free) capacity for all online volumes in the storage group, in gigabytes. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number

Table 340. Response content for a successful retrieval of storage-group summary information (continued)

Field name	Description	Data type
spaceUsed	Indicates the total amount of used (allocated) capacity for all online volumes in the storage group, in gigabytes.	Number
status	<p>The status object is an array of four string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the storage group. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	Array
storageGroupName	<p>Indicates the storage group name.</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String
storageGroupType	<p>Indicates the storage group type. The value is one of the following types:</p> <ul style="list-style-type: none"> • VIO • Pool • Dummy • Copy Target • Object • Object backup • Tape 	String

Table 340. Response content for a successful retrieval of storage-group summary information (continued)

Field name	Description	Data type
totalSpace	Indicates the total capacity that is obtained from all online volumes in the storage group, in gigabytes. Only volumes that are connected to a valid system, for storage groups in the active configuration, are included. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
updateDate	Indicates the date that the storage group definition was last changed.	String
updateTime	Indicates the time that the storage group definition was last changed.	String

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 341 on page 563](#).

Table 341. Response content for a successful retrieval of storage-group detailed information

Field name	Description	Data Type
autoBackupSystem	Indicates the name of the system on which automatic backup of the volumes in the storage group takes place.	String
autoDump	Indicates whether the volumes in this storage group can be automatically dumped or automatically backed up by using DFSMSHsm or a comparable product.	Boolean
autoDumpSystem	Indicates the name of the system on which automatic dumping of the volumes in the storage group takes place.	String
autoMigrateSystem	Indicates the name of the system on which automatic migration and space management of the volumes in this storage group takes place. If the field is blank, automatic migration or space management can take place on any system.	String
autoMigration	Indicates whether data sets on volumes in this storage group can be moved to DASD or tape by the primary space management and interval migration functions of DFSMSHsm.	Boolean

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
breakPointValue	<p>Indicates the EAV breakpoint value, which is used in determining whether a data set can be allocated in a cylinder-managed space. Valid values are 0-65.</p> <p>The breakpoint value is expressed as a number of cylinders.</p> <p>When a disk space request is this size or larger, the system gives the cylinder-managed space preference for that extent. The preference applies to each request for primary or secondary space for data sets that are eligible for the cylinder-managed space. If not enough cylinder-managed space is available, then the system uses the track-managed space or uses both areas. When the size of a disk space request is less than the breakpoint value, the system gives the track-managed space preference. If enough space is not available there, the system uses the cylinder-managed space or uses both areas.</p>	Number
breakPointValueSpecified	Indicates whether the EAV breakpoint value is defined.	Boolean
driveStartThresholdBit	Indicates the maximum number of outstanding object write requests for an optical drive in this storage group. Valid values are 0-127.	Number
dumpClassesForAD	Indicates the names of the dump classes that are assigned to this storage group.	String
EAVTrackAllocationThresholdSpecified	Indicates whether the EAV track allocation threshold is defined.	Boolean
extendSgexname	Indicates the name of another pool storage group to which data sets from the primary storage group can be extended when the primary storage group has an insufficient amount of storage space.	String
extendSgexnameLength	Indicates the length of the storage group extend name. Valid values are 0-30.	Number
extendSGSpecified	Indicates whether data sets from the primary storage group can be extended when the primary storage group has an insufficient amount of storage space.	Boolean
extendSgtcname	Indicates the name of the copy-pool backup storage-group that contains volumes that are eligible for fast replication backup versions. The storage group can use the backup storage group only if the value of targetCopySGSpecified is true.	String
extendSgtcnameLength	Indicates the length of the target copy storage group name. Valid values are 0-30.	Number
guaranteedBackupFreq	<p>Indicates the maximum number of days between backups. The property is valid only for pool storage groups. Valid values are 1-9999.</p> <p>The property is ignored if the value of guaranteedBackupFreqNoLimit is "ON".</p>	Number

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
guaranteedBackupFreqNoLimit	Indicates whether the management class specifies how often data sets are backed up. Valid values are "ON" and "OFF". If the value is "OFF", the guaranteedBackupFreq value indicates the guaranteed backup frequency. If the value is "ON", the management class defines backup frequency.	String
guaranteedBackupFreqSpecified	Indicates whether the maximum number of days between backups is defined.	Boolean
highThreshold	Indicates the maximum space limit for the DASD volumes in a pool storage group. Valid values are 1-100.	Number
highTrackThreshold	Indicates the high threshold percent for the track-managed portion of the volume. Valid values are 1-100.	Number
HSMAutoBackup	Indicates whether the data sets on the volumes in the storage group are eligible for automatic backup.	Boolean
intervalMigration	Indicates whether interval migration is defined.	Boolean
lastUser	Indicates the user ID of the user that made the last update to the storage group definition.	String
libraryName	Indicates the optical library name.	String
libraryNameLength	Indicates the optical library name length.	Number
lowThreshold	Indicates the low threshold percent for the track-managed portion of the volume. The value applies to the primary storage group. Valid values are 0-99.	Number
lowTrackThreshold	Indicates the threshold percentage of space allocation that stops migration of data sets from volumes in this storage group during interval migration. The value applies to the track-managed space of an extended address volume. Valid values are 0-99.	Number
numberOfVolumes	Indicates the number of volume entries in the storage group's volume list. The number includes predefined volume serials, which are volume entries for which no physical volume exists.	Number
oamCycleEndtime	Indicates the last hour of a window of time in which object processing can be started for this storage group. Valid values are 0-23 and "None".	String
oamCycleStarttime	Indicates the first hour of a window of time in which object processing can be started for this storage group. Valid values are 0-23 and "None".	String

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
oamFlags	<p>Indicates the OAM flags for the storage group. Valid values are 0-127. The following properties are encoded:</p> <ul style="list-style-type: none"> • X'80': Whether OAM cycle start and end times are given. The times are defined in the oamCycleStartTime and oamCycleEndTime fields. • X'40': "Volume full bit": Whether optical volumes are marked as full when they have insufficient free space. The required free space is defined in the volumeFullThresholdBit field. • X'20': "Drive start threshold bit": Whether a maximum limit is defined for the number of outstanding object write requests for an optical drive in this storage group. The maximum limit is defined in the driveStartThresholdBit field. • X'10': Whether a "mark volume full on first write failure" criteria is defined for the storage group. • X'08': Whether retention protection is enabled for an object. When retention protection is enabled, OAM prevents an object from being deleted before the object's expiration date. Additionally, OAM prevents the expiration date from being changed to an earlier date. The expiration date can be changed to a later date. • X'04': Whether deletion protection is enabled for an object. When deletion protection is enabled, OAM prevents an object from being deleted before the object's expiration date. Deletion protection differs from retention protection in that the deletion protection can be turned on and off by the installation, and deletion protection does not restrict any changes to the expiration date. • X'02': This bit is reserved. • X'01': This bit is reserved. 	Number
oamTableSI	Indicates the OAM table space identifier for this storage group in the form GROUPnn.	String
osmcSystemName	Indicates which OAM system in the sysplex completes OSMC processing for this object storage group.	String
overflowSG	If overflow designation is defined, this property indicates whether this pool storage group is designated as an overflow storage group. An overflow storage group handles periods of high demand for initial primary space allocations. Volumes in overflow storage groups are selected for primary space allocation only when all the volumes in non-overflow storage groups cannot satisfy the allocation amount without exceeding the high threshold for the storage group.	Boolean
OverflowSGSpecified	Indicates whether overflow designation is defined.	Boolean

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
primarySpaceManagement	<p>Indicates whether data sets are eligible for primary space management only. If the value is TRUE, then the following eligibility applies:</p> <ul style="list-style-type: none"> • Data sets are eligible for primary space management (daily migration) only. • Data sets are not eligible for interval migration (hourly migration) even if Setsys Intervalmigration is specified in DFSMSHsm. <p>For space management processing, DFSMSHsm processes storage groups in the order of their processing priority. Storage groups that are migration targets for class transition processing require free space for the transitions. Those storage groups must have a relatively high processing priority.</p>	Boolean
processingPriority	<p>Indicates the space management processing priority. Valid values are 1-100.</p> <p>For space management processing, DFSMSHsm processes storage groups in the order of their processing priority. Storage groups that are migration targets for class transition processing require free space for the transitions. Those storage groups must have a relatively high processing priority.</p>	Number
processingPrioritySpecified	Indicates whether processing priority is specified.	Boolean
SMADDataSpecified	The SMS IPCS verb exit (SMSDATA) is intended for the use of diagnosticians who are working with the IBM Support Center to resolve an SMS-related problem.	Boolean
spaceAvailable	Indicates the total amount of available (free) capacity for all online volumes in the storage group, in gigabytes. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
spaceUsed	Indicates the total amount of used (allocated) capacity for all online volumes in the storage group, in gigabytes.	Number

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
status	<p>The status object is an array of four string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the storage group. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	Array
storageGroupName	<p>Indicates the storage group name.</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String
storageGroupType	<p>Indicates the storage group type. The value is one of the following types:</p> <ul style="list-style-type: none"> • VIO • Pool • Dummy • Copy Target • Object • Object backup • Tape 	String

Table 341. Response content for a successful retrieval of storage-group detailed information (continued)

Field name	Description	Data Type
targetCopySGSpecified	Indicates whether the storage group can use a copy-pool backup storage-group for fast-replication backup versions. When the value is true, the storage group that is defined in extendSGtcname is valid.	Boolean
thresholdsSpecified	Indicates whether space thresholds are specified.	Boolean
tmsTotalSpaceAlert	Indicates the point at which storage use as a percentage of total storage exceeds the acceptable level. The value applies among the storage group's TAPE type storage.	Number
totalSpace	Indicates the total capacity that is obtained from all online volumes in the storage group, in gigabytes. Only volumes that are connected to a valid system, for storage groups in the active configuration, are included. Volumes are included even if they are ineligible for data set allocation because of their SMS status.	Number
totalSpaceAlert	Indicates the point at which storage use for the storage group, as a percentage of total storage, exceeds the acceptable level. The value applies among the storage group's TAPE type storage. Valid values are 0-99.	Number
updateDate	Indicates the date that the storage group definition was last changed.	String
updateTime	Indicates the time that the storage group definition was last changed.	String
vioMaxDss	Indicates the largest size of a virtual input/output (VIO) data set that you can create for this storage group, in kilobytes. You cannot allocate data sets that exceed this size in this Storage Group. This value applies to VIO storage groups only.	Number
vioUnitType	Indicates the type of physical device that is simulated by the storage group. At least one unit of the device type must be physically connected to each system that has access to the storage group. This value applies to VIO storage groups only.	String
volumeFullThresholdBit	Indicates the number of free sectors that are required for an optical volume in the storage group. When the number of free sectors in a volume becomes less than the threshold, the Object Access Method marks the optical volume as full. This value is valid only for the OBJECT and OBJECT BACKUP storage group types. Valid values are 0-127.	Number

Example HTTP interaction

1. The example in Figure 257 on page 569 shows a request to get a list of storage groups.

```
GET zosmf/storage/rest/v1/storagegroups/DOMUS?detail-data=Y
```

Figure 257. Sample request to get a storage-group definition

The following is the response body for the request.

```
[
{
  "oamFlags": "",
  "spaceAvailable": 0,
  "lastUser": "KNMORT",
  "vioUnitType": "",
  "oamTableSI": "",
  "processingPrioritySpecified": false,
  "highTrackThreshold": 0,
  "extendSGSpecified": true,
  "autoBackupSystem": "",
  "overflowSGSpecified": false,
  "lowTrackThreshold": 0,
  "extendSgexnameLength": 0,
  "storageGroupName": "DOMUS",
  "totalSpaceAlert": 0,
  "overflowSG": 0,
  "autoMigrateSystem": "",
  "SMADDataSpecified": false,
  "extendSgtcname": "",
  "autoDumpSystem": "",
  "autoMigration": true,
  "primarySpaceManagement": false,
  "breakPointValueSpecified": false,
  "highThreshold": 99,
  "HSMAutoBackup": false,
  "breakPointValue": 0,
  "status": [
    {
      "sysType": "1",
      "confirmedSmsStatus": "DISABLED/ALL",
      "sysName": "S1I",
      "requestedSystemStatus": "ENABLED"
    },
    ...
    {
      "sysType": "2",
      "confirmedSmsStatus": "DISABLED/ALL",
      "sysName": "XCFLOCAL",
      "requestedSystemStatus": "ENABLED"
    }
  ],
  "libraryName": "",
  "updateDate": "2003/08/07",
  "intervalMigration": false,
  "spaceUsed": 0,
  "oamCycleStartTime": 0,
  "description": "STORAGE GROUP FOR DOMINO WORKLOAD",
  "storageGroupType": "POOL",
  "osmcSystemName": "",
  "vioMaxDss": 0,
  "processingPriority": 0,
  "guaranteedBackupFreqSpecified": true,
  "driveStartThresholdBit": 0,
  "extendSgtcnameLength": 0,
  "EAVTrackAllocThresholdSpecified": false,
  "tmsTotalSpaceAlert": 0,
  "lowThreshold": 1,
  "numberOfVolumes": 15****,
  "guaranteedBackupFreq": 0,
  "updateTime": "11:19",
  "totalSpace": 0,
  "volumeFullThresholdBit": 0,
  "dumpClassesForAD": "",
  "targetCopySGSpecified": false,
  "thresholdsSpecified": true,
  "subsystemUse2": 29648,
  "subsystemUse1": 15,
  "extendSgexname": "",
  "autoDump": false,
  "libraryNameLength": 0,
  "oamCycleEndTime": 0
}
]
```

Figure 258. Sample response body

Get a list of volumes

This operation retrieves a list of volumes that match the specified query parameters. Each returned record contains properties for one volume.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/volumes
```

Query parameters

You can include query parameters on this request to filter volumes. See [Table 342 on page 571](#) for a description of the query parameters.

Table 342. Query parameters for retrieving a list of volumes

Query Parameter	Description	Required or Optional	Type
filter	Specifies a volume serial number. The value must meet the following criteria: <ul style="list-style-type: none">• Contains 1-6 alphanumeric characters• The asterisk (*) and percent (%) wildcards are allowed.<ul style="list-style-type: none">– Asterisk matches any number of characters.– Percent matches one character.– Asterisk and percent wildcards can be used together.– You can use more than one asterisk and more than one percent wildcard.	Optional	String
stg-name	Specifies the name of a storage-group that is associated with the volume. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none">• Contains 1-8 alphanumeric characters• Begins with an alphabetic or national character. National characters are \$, #, and @.• Wildcards are not allowed.	Optional	String
offset	Specifies the offset for the list of volumes. The request retrieves volumes starting from this offset value. For example, an offset of 50 means retrieve from the 51st position in the list and onward. The valid range of offset values is 0 - 2147483647. The offset cannot be larger than or equal to the number of volumes. The offset cannot be used with the filter parameter in the same request. The default value for offset is 0.	Optional	Number

Table 342. Query parameters for retrieving a list of volumes (continued)

Query Parameter	Description	Required or Optional	Type
limit	<p>Specifies the maximum number of volumes that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>You can use the limit parameter with the offset parameter. For example, if you specify offset = 1 and limit = 3, the request retrieves the second, third, and fourth volumes from the result set.</p> <p>Observe the following usage notes:</p> <ul style="list-style-type: none"> • If you specify limit = 0, the request retrieves all of the queried volumes from the offset value. • If the offset plus limit is a larger value than the number of queried volumes, the request retrieves only the number of the queried volumes, minus the offset. • If the offset plus limit is a smaller value than the number of queried volumes, the request retrieves only the limit number of volumes, starting from the offset. <p>The limit cannot be used with the filter parameter in the same request.</p> <p>The default value for limit is 0.</p>	Optional	Number

Description

On successful completion, the request returns a list of volume records. The returned content includes a record for each volume that matches the specified query parameters. For a description of the response content, see [“Response content” on page 572](#).

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains an array of JSON storage group documents that match the specifications. Each document contains the information in [“Response content” on page 572](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content

Successful queries return information in fields that are described in [Table 343 on page 573](#).

Table 343. Response content for a successful retrieval of volume information

Field name	Description	Data Type
freeSpace	Indicates the total amount of free space, in megabytes.	Number
freeSpaceForTMS	Indicates the total amount of free space in megabytes that is described by the track-managed space on the volume.	Number
fullVolumeLastUsed	Indicates the percentage of total space that is in use. Possible values are 0-100.	Number
largestFreeExtent	Indicates the largest free extent, in megabytes.	Number
largestFreeExtentForTMS	Indicates the largest free space extent that is described by the track-managed space on the volume, in megabytes.	Number
lastUser	Indicates the user ID of the user that made the last update to the volume definition.	String

Table 343. Response content for a successful retrieval of volume information (continued)

Field name	Description	Data Type
status	<p>The status object is an array of five string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the volume. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>mvsSystemStatus Indicates the volume status that is set by the MVS environment. The following values are valid:</p> <ul style="list-style-type: none"> • ONLINE • OFFLINE • PENDING OFFLINE • BOXED • NOT READY <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	String
storageGroupName	<p>Indicates the name of the storage group that is associated with the volume:</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String

Table 343. Response content for a successful retrieval of volume information (continued)

Field name	Description	Data Type
storageGroupStatus	Indicates the status of the storage group that is associated with the volume. The value is one of the following statuses: <ul style="list-style-type: none"> • NONE • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW 	String
totalCapacity	Indicates the total capacity of the volume, in megabytes.	Number
totalCapacityForTMS	Indicates the total capacity of the track-managed space on the volume, in megabytes.	Number
trackRegionLastUsed	Indicates the percentage of total track-managed space that is in use. Possible values are 0-100.	Number
updateDate	Indicates the date that the volume definition was last changed.	String
updateLevel	Indicates the volume update level.	String
updateTime	Indicates the time that the volume definition was last changed.	String
volumeSerial	Indicates the volume serial number.	String

Example HTTP interaction

1. The example in [Figure 259](#) on page 575 shows a request to get a list of volumes.

```
GET zosmf/storage/rest/v1/volumes?filter=DOMUSB
```

Figure 259. Sample request to get a list of volumes

The following code is the response body for the request.

```
[
  {
    "storageGroupName": "DOMUS",
    "lastUser": "KNMORT",
    "updateDate": "2003/08/07",
    "totalCapacityForTMS": 0,
    "volumeSerial": "DOMUSB",
    "updateLevel": 0,
    "freeSpace": 0,
    "storageGroupStatus": "ENABLED",
    "largestFreeExtentForTMS": 0,
    "updateTime": "11:22",
    "freeSpaceForTMS": 0,
    "fullVolumeLastUsed": 0,
    "trackRegionLastUsed": 0,
    "totalCapacity": 0,
    "largestFreeExtent": 0,
    "status": [
      {
        "confirmedSmsStatus": "DISABLED/ALL",
        "sysType": "1",
        "requestedSystemStatus": "NONE",
        "sysName": "S1I",
        "mvsSystemStatus": "OFFLINE"
      },
      ...
      {
        "confirmedSmsStatus": "NONE",
        "sysType": "2",
        "requestedSystemStatus": "NONE",
        "sysName": "XCFLOCAL",
        "mvsSystemStatus": "OFFLINE"
      }
    ]
  }
]
```

Figure 260. Sample response body

Get a volume definition

This operation retrieves a volume definition. The response content contains properties of the volume.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/volumes/<volume-ser>
```

Query parameters

You request must include the volume serial number, which is specified as a query parameter.

Table 344. Query parameter for retrieving a volume definition

Query Parameter	Description	Required or Optional	Type
volume-ser	Specifies a volume serial number. The value must be a complete serial number, 1 - 6 characters, no wildcard characters. Partial volume numbers are not matched. The value is not case-sensitive.	Required	String

Description

On successful completion, the query returns a record of the properties of the volume. For a description of the response content, see [“Response content” on page 577](#).

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains a JSON volume document. The document contains the information in [“Response content” on page 577](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content

Successful queries return information in fields that are described in [Table 345 on page 577](#).

Table 345. Response content for a successful retrieval of volume information		
Field name	Description	Data Type
freeSpace	Indicates the total amount of free space, in megabytes.	Number
freeSpaceForTMS	Indicates the total amount of free space in megabytes that is described by the track-managed space on the volume.	Number
fullVolumeLastUsed	Indicates the percentage of total space that is in use. Possible values are 0-100.	Number
largestFreeExtent	Indicates the largest free extent, in megabytes.	Number
largestFreeExtentForTMS	Indicates the largest free space extent that is described by the track-managed space on the volume, in megabytes.	Number
lastUser	Indicates the user ID of the user that made the last update to the volume definition.	String

Table 345. Response content for a successful retrieval of volume information (continued)

Field name	Description	Data Type
status	<p>The status object is an array of five string elements:</p> <p>sysName Indicates the system name or the sysplex name.</p> <p>sysType Indicates whether the named system is a single system or a sysplex. One of three numeric values indicates the type:</p> <p>0 Type is not defined.</p> <p>1 Type is system.</p> <p>2 Type is sysplex.</p> <p>requestedSystemStatus Indicates the requested status of the volume. The requested status is the status that is defined by a user with appropriate authority. The requested status might not be the same as the confirmed SMS status. The following values are valid:</p> <ul style="list-style-type: none"> • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW • NONE <p>mvsSystemStatus Indicates the volume status that is set by the MVS environment. The following values are valid:</p> <ul style="list-style-type: none"> • ONLINE • OFFLINE • PENDING OFFLINE • BOXED • NOT READY <p>confirmedSmsStatus If the storage group status slot is not defined in the base configuration, the value is "NONE". The value is the same as the requested system status.</p>	String
storageGroupName	<p>Indicates the name of the storage group that is associated with the volume:</p> <ul style="list-style-type: none"> • The property contains up to 8 characters of the storage group name. 	String

Table 345. Response content for a successful retrieval of volume information (continued)

Field name	Description	Data Type
storageGroupStatus	Indicates the status of the storage group that is associated with the volume. The value is one of the following statuses: <ul style="list-style-type: none"> • NONE • ENABLED • QUIESCED/ALL • QUIESCED/NEW • DISABLED/ALL • DISABLED/NEW 	String
totalCapacity	Indicates the total capacity of the volume, in megabytes.	Number
totalCapacityForTMS	Indicates the total capacity of the track-managed space on the volume, in megabytes.	Number
trackRegionLastUsed	Indicates the percentage of total track-managed space that is in use. Possible values are 0-100.	Number
updateDate	Indicates the date that the volume definition was last changed.	String
updateLevel	Indicates the volume update level.	String
updateTime	Indicates the time that the volume definition was last changed.	String
volumeSerial	Indicates the volume serial number.	String

Example HTTP interaction

1. The example in [Figure 261](#) on page 579 shows a request to get a volume definition.

```
GET zosmf/storage/rest/v1/volumes/DOMUSB
```

Figure 261. Sample request to get a volume definition

The following is the response body for the request.

```
[
  {
    "storageGroupName": "DOMUS",
    "lastUser": "KNMORT",
    "updateDate": "2003/08/07",
    "totalCapacityForTMS": 0,
    "volumeSerial": "DOMUSB",
    "updateLevel": 0,
    "freeSpace": 0,
    "storageGroupStatus": "ENABLED",
    "largestFreeExtentForTMS": 0,
    "updateTime": "11:22",
    "freeSpaceForTMS": 0,
    "fullVolumeLastUsed": 0,
    "trackRegionLastUsed": 0,
    "totalCapacity": 0,
    "largestFreeExtent": 0,
    "status": [
      {
        "confirmedSmsStatus": "DISABLED/ALL",
        "sysType": "1",
        "requestedSystemStatus": "NONE",
        "sysName": "S1I",
        "mvsSystemStatus": "OFFLINE"
      },
      ...
      {
        "confirmedSmsStatus": "NONE",
        "sysType": "2",
        "requestedSystemStatus": "NONE",
        "sysName": "XCFLOCAL",
        "mvsSystemStatus": "OFFLINE"
      }
    ]
  }
]
```

Figure 262. Sample response body

Get a list of data classes

This operation retrieves a list of data class records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all data classes that match the query parameters.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/dataclasses
```

Query parameters

You can include query parameters on this request to filter data classes and determine the level of detail in the returned content. For descriptions of the query parameters, see [Table 346 on page 581](#).

Table 346. Query parameters for retrieving a list of data classes

Query Parameter	Description	Required or Optional	Type
filter	<p>Specifies a data class name.</p> <p>The value is not case-sensitive.</p> <p>The value must meet the following criteria:</p> <ul style="list-style-type: none"> • Contains 1-8 alphanumeric characters. • Begins with an alphabetic or national character. National characters are \$, #, and @. • The asterisk (*) and percent (%) wildcards are allowed. <ul style="list-style-type: none"> – Asterisk matches any number of characters. – Percent matches one character. – Asterisk and percent wildcards can be used together. – You can use more than one asterisk and more than one percent wildcard. 	Optional	String
detail-data	<p>Specifies whether the response contains detailed information.</p> <p>Valid values are Y and N.</p> <p>A Y value requests detailed information. A N value requests summary information.</p> <p>The default value is N. (N is used if Y is not specified.)</p>	Optional	String
offset	<p>Specifies the offset for the list of data classes.</p> <p>The request retrieves data classes starting from this offset value. For example, an offset of 50 means retrieve from the 51st position in the list and onward. The valid range of offset values is 0 - 2147483647.</p> <p>The offset cannot be larger than or equal to the number of data classes.</p> <p>The offset cannot be used with the filter parameter in the same request.</p> <p>The default value for offset is 0.</p>	Optional	Number

Table 346. Query parameters for retrieving a list of data classes (continued)

Query Parameter	Description	Required or Optional	Type
limit	<p>Specifies the maximum number of data classes that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>You can use the limit parameter with the offset parameter. For example, if you specify offset = 1 and limit = 3, the request returns the second, third, and fourth data classes from the result set.</p> <p>Observe the following usage notes:</p> <ul style="list-style-type: none"> • If you specify limit = 0, the request returns all of the queried data classes from the offset value. • If the offset plus limit is a larger value than the number of queried data classes, the request returns only the number of the queried data classes, minus the offset. • If the offset plus limit is a smaller value than the number of queried data classes, the request returns only the limit number of data classes, starting from the offset. <p>The limit cannot be used with the filter parameter in the same request.</p> <p>The default value for limit is 0.</p>	Optional	Number

Description

On successful completion, the request returns a list of data class records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all data classes that match the query parameters. For descriptions of the query parameters that you can specify on the request, see [Table 346 on page 581](#).

For descriptions of the response content, see [“Response content: Summary record” on page 583](#) and [“Response content: Detailed record” on page 595](#).

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains an array of JSON data classes documents that match the specifications. If you do not request detailed information, each document contains only the information in [“Response content: Summary record” on page 583](#). If you request detailed information, each document contains the information in [“Response content: Detailed record” on page 595](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 347](#) on page 583.

Table 347. Response content for a successful retrieval of data classes summary information	
Field name	Description
dataClassName	Data class name, consisting of 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.
lastUser	User ID of the user that made the last update to the data class definition.
updateDate	Date that the storage class definition was last changed.
updateTime	Time that the storage class definition was last changed.
recorg	<p>Specifies how VSAM data sets allocated by a data class are organized.</p> <p>Possible values include:</p> <p>ES VSAM Entry Sequenced Data Set</p> <p>KS VSAM Keyed Sequential Data Set</p> <p>LS VSAM Linear Space Data Set</p> <p>RR VSAM Relative Record Data Set</p> <p>Blank If the value has not been specified, the value will be blank.</p>
recfm	<p>Specifies the record format and type of carriage control assigned to non-VSAM data sets.</p> <p>Possible values include:</p> <p>A ANSI carriage control</p> <p>B Blocked format</p> <p>F Fixed format</p> <p>M Machine carriage control</p> <p>S Standard (if format is fixed) or Spanned (if format is variable)</p> <p>U Undefined format</p> <p>V Variable format</p> <p>Blank If the value has not been specified, the value will be blank.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
vsamExtendedAddressing	<p>Specifies whether extended addressability is provided. Extended addressability provides data sets with addressability of more than 4 gigabytes.</p> <p>Possible values include:</p> <p>TRUE Provides extended addressability.</p> <p>FALSE Does not provide extended addressability.</p>
spaceConstraintRelief	<p>Specifies whether space constraint relief method was selected or not during data class define/alter.</p> <p>Possible values include:</p> <p>TRUE Re-allocation methods are to be used by SMS in case of allocation failure.</p> <p>FALSE No attempt of retry are to be used by SMS in case of allocation failure.</p>
vsamKeyOffset	<p>Distance (in bytes) from the start of the record to the start of the key field. For non-VSAM data sets, KEYOFF is ignored.</p> <p>Possible values: 0 - 32760. If the value has not been specified, the value will be blank.</p>
vsamKeyLength	<p>Distance (in bytes) from the start of the record to the start of the key field. For non-VSAM data sets, KEYOFF is ignored.</p> <p>Possible values: 0 - 255 for non-VSAM data sets and 1 - 255 for key-sequenced VSAM data sets. If the value has not been specified, the value will be blank.</p>
primarySpaceAmount	<p>Amount of space the data class initially allocated for a data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>
secondarySpaceAmount	<p>Additional space that can be allocated for a data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>
directoryBlocks	<p>Number of blocks allocated for the directory of a partitioned data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
avgrec	<p>Specifies whether the data class allocates space in bytes, kilobytes, or megabytes.</p> <p>Possible values include:</p> <p>K Space is allocated in kilobytes.</p> <p>M Space is allocated in megabytes.</p> <p>U Space is allocated in bytes.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
reduceSpaceUpTo	<p>Specifies the amount in % to be used by SMS for reducing the requested space quantity during data set allocation failure.</p> <p>Possible values: 0 - 99. If the value has not been specified, the value will be blank.</p>
vsamRecordAccessBias	<p>Allows the system to acquire and choose the buffering algorithms.</p> <p>Possible values include:</p> <p>SYSTEM Allows the system to choose the number of buffers and the buffering algorithms for the VSAM data set.</p> <p>USER Number of buffers and buffering algorithms will be based on user specified (or defaulted) values and current algorithms.</p> <p>DO Indicates system managed buffering with direct optimization.</p> <p>DW Indicates system managed buffering with sequential optimization.</p> <p>SO Indicates system managed buffering with sequential optimization.</p> <p>SW Indicates system managed buffering weighted for sequential processing.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
dynamicVolumeCount	<p>Maximum number of volumes that DFSMS can dynamically add to an SMS managed data set.</p> <p>Possible values: 1 - 59. If the value has not been specified, the value will be blank.</p>
avgValue	<p>Multiplication factor used in determining allocated space.</p> <p>Possible values: 0 - 65535. If the value has not been specified, the value will be blank.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
spaceOverrideFlag	<p>Specifies whether the data class space attributes will override the space attributes from other sources, such as JCL.</p> <p>Possible values:</p> <p>TRUE Data class attributes will override others.</p> <p>FALSE Data class attributes can be overridden.</p>
forceSystemDeterminedBlockSize	<p>Specifies whether the system will ignore a user-specified block size if no program opens the data set for writing while it still is allocated. This prevents the user from overriding a system-determined block size.</p> <p>Possible values:</p> <p>TRUE If no program opens the data set for writing while the new data set still is allocated, then the system will discard a BLKSIZE value coded by the user. The system will attempt to determine an optimal block size. If a program opens the data set for output while it still is allocated, then the user-specified BLKSIZE will take effect and override a system-determined block size.</p> <p>FALSE If the user specifies a BLKSIZE value, it will take effect and override a system-determined block size. This is the normal way for the system to run.</p>
eattr	<p>Specifies whether the data set can support extended attributes DSCBs.</p> <p>Possible values:</p> <p>NO No extended attribute DSCBs. Only a format 1 DSCB will be created for this data set even if the data set is allocated on a volume that supports extended attribute DSCBs.</p> <p>OPT Extended attribute DSCBs are optional. Format 8 and 9 DSCBs will be created for this data set when allocated on a volume that supports extended attribute DSCBs. Otherwise, a format 1 DSCB will be created.</p> <p>Blank Not specified and defaults are used.</p> <p>For non-VSAM data sets, the EATTR value used by the system is equivalent to NO. For VSAM data sets, the EATTR value used by the system is equivalent to OPT.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
vsamSMBRMode31	<p>Specifies where the buffers and control blocks are to reside.</p> <p>Possible values:</p> <p>ALL Buffers and control blocks reside above the line.</p> <p>BUFF Buffers only reside above the line.</p> <p>CB Control blocks reside above the line.</p> <p>NONE Buffers and control blocks reside below the line.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
recordLength	<p>Logical record length (in bytes) used when allocating data sets in this data class. The value displayed is the length of fixed-length records or the maximum length of variable-length records.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 1 - 32761 or blank for non-VSAM data sets • 1 - 32760 or blank for VSAM data sets. • If the value has not been specified, the value will be blank.
ciSize	<p>Number of bytes allocated for each control interval in the data portion, not the index portion, of a data set. To allow for overhead processing, the CISE value will be at least seven bytes greater than maximum record size.</p> <p>Possible values: 1 - 32768. If the value has not been specified, the value will be blank.</p>
ciFreeSpace	<p>Specifies what percentage of each control interval in a key-sequenced VSAM data set should be set aside as free space. VSAM uses the space to lengthen or insert records, as needed.</p> <p>Possible values: 0 - 100. If the value has not been specified, the value will be blank.</p>
caFreeSpace	<p>Specifies what percentage of each control area in a key-sequenced VSAM data set should be set aside as free space. VSAM uses the space to lengthen or insert records, as needed.</p> <p>Possible values: 0 - 100. If not specified, the value will be blank.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
vsamXSystemShareOptions	<p>Specifies how a VSAM data set can be shared among systems.</p> <p>Possible values:</p> <p>3 All users can read and update the data set. VSAM does not protect the data set.</p> <p>4 All users can read and update the data set. VSAM monitors access of data sets in order to prevent lost, damaged, or altered data.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
vsamXRegionShareOptions	<p>Specifies how a VSAM data set can be shared among regions of one system or across regions of multiple systems.</p> <p>Possible values:</p> <p>1 All users can read the data set OR one user can update it.</p> <p>2 All users can read the data set AND one user can update it.</p> <p>3 All users can read and update the data set. VSAM does not protect the data set.</p> <p>4 All users can read and update the data set. VSAM helps prevent lost, damaged or altered data.</p> <p>Blank If the value has not been specified, the system will use the default value of primary.</p>
additionalVolumeAMT	<p>Type of allocation amount when a VSAM data set in extended format begins allocation on subsequent new volumes.</p> <p>Possible values:</p> <p>PRIMARY Primary allocation amount has been requested.</p> <p>SECONDARY Secondary allocation amount has been requested.</p> <p>Blank If the value has not been specified, the system will use the default value of primary.</p>
volumeCount	<p>Maximum number of volumes that can be used to store your data set.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 1 - 59 (DASD) • 1 - 255 (Tape) • If the value has not been specified, the value will be blank.

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
dsnType	<p>Data set name type.</p> <p>Supported types include:</p> <ul style="list-style-type: none"> • PDS • LIBRARY • HFS • EXTENDED PREFERRED • EXTENDED REQUIRED • LARGE • If the value has not been specified, the value will be blank.
retpdYear	<p>Default retention period or expiration date of data sets in this data class. Data sets are deleted or archived one day after the retention period or on the expiration date. Each data class can have a retention period or an expiration date (or neither), but not both.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • If retpdYear is blank, the expiration date is not specified. • If retpdAbsoluteDayOfYear is blank, the retention period is not specified. • If the retpdYear and retpdAbsoluteDayOfYear are not blank, the expiration date is specified. • If retpdYear is blank and retpdAbsoluteDayOfYear is not blank, retention period is specified. • Year range: 1900-2155.
retpdAbsoluteDayOfYear	<p>Absolute day of year. Data sets expire in the number of days displayed (0 - 93000).</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
compactionType	<p>This property shows whether tape volumes or DASD data sets associated with this data class are compacted or compressed. Compaction and compression increase storage capacity.</p> <p>Possible values:</p> <p>YES Extended format data sets are compressed and tape volumes are compacted. The type of DASD compression depends on the COMPRESS option in IGDSMSxx. Tape volumes are compacted unless overridden by the user through JCL/dynamic allocation.</p> <p>NO Data sets are not compressed. Tape volumes are not compacted, unless requested by the user on JCL/dynamic allocation.</p> <p>TCOM Extended format data sets are compressed using tailored dictionaries overriding SYS1.PARMLIB.</p> <p>GEN Extended format data sets are compressed using generic dictionaries overriding SYS1.PARMLIB.</p> <p>ZR The system will fail the allocation request if the zEDC function is not supported by the system or the minimum allocation amount requirement is not met.</p> <p>ZP The system will not fail the allocation request but rather create either a tailored compressed data set if the zEDC function is not supported by the system or create a non-compressed extended format data set if the minimum allocation amount requirement is not met.</p> <p>Blank Data sets are not compressed. Tape volumes may be compacted depending on what was specified by the user on JCL/dynamic allocation, the installation with the COMPACT option in parmlib member DEVSUPxx, or the allocated hardware model.</p> <p>-- Unexpected error.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
mediaType	<p>Media type.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • MEDIA1 - Cartridge system • MEDIA2 - Enhanced capacity cartridge system tape • MEDIA3 - ½ inch / 320 meter particle media (3590) • MEDIA4 - Double length (3590) • MEDIA5 - 3592 module J, 3592 Model E05 • MEDIA5 - 3592 module JW, 3592 Model E05 • MEDIA5 - 3592 module JV, 3592 Model E05 • MEDIA5 - 3592 module JR, 3592 Model E05 • MEDIA9 - 3592 model E05 • MEDIA10 - 3592 model E05 • If the value has not been specified, the value will be blank.
recordingTechnology	<p>Recording technology type.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 18 track • 36 track • 128 track • 256 track • 384 track • EFMT1 recording technology • EFMT2 recording technology • EEFMT2 recording technology • EFMT3 recording technology • EEFMT3 recording technology • EFMT4 recording technology • EEFMT4 recording technology • Blank, if not specified
vsamExtentionReuse	<p>Specifies whether you can open the cluster repeatedly as a new cluster.</p> <p>Possible values:</p> <p>TRUE Cluster is reusable.</p> <p>FALSE Cluster is not reusable.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
initialLoad	<p>Specifies whether the storage allocated to the data component was preformatted before records were inserted during initial load.</p> <p>Possible values:</p> <p>SPEED Indicates the data component's space was not preformatted.</p> <p>RECOVERY Indicates the data component's space was preformatted.</p>
blockSizeLimit	<p>Specifies the largest value that the system can determine for a data set block size when a program opens a dataset for output and no value is available. The system determines a data set block size that is appropriate for the media type when the data set is sequential, the record format is fixed or variable, and the logical record length is known. The BLKSZLIM keyword on the DD statement overrides this value. Limit the block size when a system or application that may read the tape does not support large blocks. Large blocks generally are more efficient.</p> <p>The actual block size determined by the system may be less than the value displayed.</p> <p>Possible values: 32760 - 2147483648. If the value has not been specified, the value will be blank.</p>
extentConstraintRemoval	<p>Specifies whether a VSAM data set is allowed to go beyond the 255 extents limit.</p> <p>Possible values:</p> <p>TRUE Remove the 255 extents limit.</p> <p>FALSE Keep the limit of 255 extents.</p>
logReplicate	<p>Specifies whether the data set will be eligible for replication.</p> <p>Possible values:</p> <p>TRUE Data set is eligible for VSAM replication.</p> <p>FALSE Data set is not eligible for VSAM replication.</p>
gsSpaceReduction	<p>Specifies whether space reduction on guaranteed space allocations is permitted or not.</p> <p>Possible values:</p> <p>TRUE Space reduction on guaranteed space allocations is permitted.</p> <p>FALSE Space reduction on guaranteed space allocations is not permitted.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
bwoType	<p>Backup While Open (BWO) type.</p> <p>Possible values:</p> <p>TYPECICS BWO processing for CICS VSAM file control data sets is permitted.</p> <p>TYPEIMS BWO processing for IMS VSAM data sets is to be used.</p> <p>NO BWO does not apply to the cluster.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
recordSpansCIAbility	<p>Specifies whether the data record is allowed to cross control interval boundaries.</p> <p>Possible values:</p> <p>SPANNED Record is contained in more than one control interval.</p> <p>NONSPANNED Record is contained in one control interval.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
frlog	<p>Specifies whether to do the logging for changes to RCC forward recovery log stream.</p> <p>Possible values:</p> <p>ALL Tells VSAM to log both forward and backward recovery logging.</p> <p>NONE No logging in effect.</p> <p>REDO Tells VSAM to do forward recovery logging.</p> <p>UNDO Tells VSAM to do backward recovery logging.</p> <p>Blank Attribute not specified.</p>
logStreamID	Specifies the name of the recovery logstream.

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
rlsCFCacheValue	<p>Specifies that VSAM RLS data with greater than 4K CI's defined to DFSMS CF cache structures can be cached.</p> <p>Possible values:</p> <p>A - ALL Indicates that all of the data is cached for the sphere.</p> <p>N - NONE Indicates that none of the data will be cached.</p> <p>U - UPDATESONLY Indicates that only updated CI's will be cached.</p> <p>D - DIRONLY Indicates that the directory only will be cached.</p>
rlsAbove2GBBar	<p>Specifies whether virtual storage for RLS can be above the 2 Gigabyte bar.</p> <p>Possible values:</p> <p>TRUE Place buffers above the bar in the SMSVSAM address space.</p> <p>FALSE Do not place any data in storage located above the bar.</p>
caReclaim	<p>Specifies whether the DASD space for empty CAs may be reused on z/OS V1.12 or later systems.</p> <p>Possible values:</p> <p>TRUE Free CAs are reused.</p> <p>FALSE Free CAs are not reused.</p>
scalingOption	<p>Tape Support use.</p> <p>Possible values:</p> <p>TRUE Scale to optimal performance</p> <p>FALSE To use the full capacity (no scaling).</p>
segmentingOption	<p>Tape Support use.</p> <p>Possible values:</p> <p>TRUE Enable segmentation format.</p> <p>FALSE No segmentation.</p>
unitForSystemManagedBufferValue	<p>Specifies unit in KB or MB for VSP value.</p>

Table 347. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
systemManagedBuffer	Amount of virtual storage for SMB Direct Access Bias obtained for buffers when opening the data set. Possible values: <ul style="list-style-type: none"> • 1KB - 2048000KB • IMB - 2048MB
dataSetKeyLabelName	Specifies the label for the encryption key used by the access methods. Possible values: 1 - 52 characters, typically containing alphanumeric, national or special characters with some additional characters also being allowed. If DASD DATA SET KEY LABEL length is more than 52 characters, the first 49 characters followed by '...' are displayed. The rest of DASD DATA SET KEY LABEL is truncated.

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 347 on page 583](#) and [Table 348 on page 595](#).

Table 348. Response content for a successful retrieval of data classes detailed information

Field name	Description
dataClassNameLength	Specifies data class name length.
description	Specifies description of data class.
blockedOrUnblocked	AMS use only. Specifies blocked or unblocked. Possible values: <ul style="list-style-type: none"> • TRUE (blocked) • FALSE (unblocked/null)
standardOrSpanned	AMS use only. Specifies standard or spanned. Possible values: <ul style="list-style-type: none"> • TRUE (standard) • FALSE (spanned)
indexOptionImbed	Specifies VSAM data set properties index option - imbed Possible values: True (IMBED) or False (No)
indexOptionReplicate	Specifies VSAM data set properties index option - replicate Possible values: TRUE (replicate) and FALSE (do not replicate)
overrideJCLSpecified	Indicates whether override JCL is specified. Possible values: True or False

Table 348. Response content for a successful retrieval of data classes detailed information (continued)

Field name	Description
sphereRecoverability	Indicates whether sphereRecoverability is specified. Possible values: 0 Not specified 1 Non-recoverable sphere 2 UNDO 3 ALL 4 REDO
logStreamIDLength	Indicates logStreamID length.
dataSetKeyLabelLength	Indicates data set key label length.
keyLabel1Length	Indicates key label 1 length.
keyLabel1Name	Indicates key label 1 name.
keyCode1	Indicates key code 1.
keyLabel2Length	Indicates key label 2 length.
keyLabel2Name	Indicates key label 2 name.
keyCode2	Indicates key code 2.

Example HTTP interaction

1. The following example shows a request to retrieve a list of all data classes on the system with summary information about each data class:

```
GET /zosmf/storage/rest/v1/dataclasses
```

Figure 263. Sample request to get a list of data classes with summary information about each data class

The following is the response body for the request.

```
[
  {
    "lastUser": "KNMORT",
    "recordLength": "",
    "caFreeSpace": "",
    "logStreamID": "",
    "scalingOption": false,
    "retpdAbsoluteDayOfYear": "",
    "spaceOverrideFlag": false,
    "directoryBlocks": "",
    "spaceConstraintRelief": false,
    "recorg": "",
    "secondarySpaceAmount": "",
    "extentConstraintRemoval": false,
    "systemManagedBuffer": "",
    "unitForSystemManagedBufferValue": "MB",
    "additionalVolumeAMT": "",
    "logReplicate": false,
    "avgValue": "",
    "gsSpaceReduction": false,
    "recfm": "",
    "vsamXRegionShareOptions": "",
    "mediaType": "",
    "frlog": "",
    "vsamXSystemShareOptions": "",
    "caReclaim": true,
    "vsamExtentionReuse": false,
    "ciFreeSpace": "",
    "vsamKeyLength": "",
    "dataSetKeyLabelName": "",
    "reduceSpaceUpTo": "",
    "updateDate": "2003/06/27",
    "rlsAbove2GBBar": false,
    "segmentingOption": false,
    "initialLoad": "RECOVERY",
    "avgrec": "",
    "recordSpansCIAbility": "",
    "primarySpaceAmount": "",
    "dataClassName": "ATL",
    "recordingTechnology": "36 track",
    "vsamKeyOffset": "",
    "ciSize": "",
    "volumeCount": 1,
    "vsamSMBRMode31": "",
    "retpdYear": "",
    "bwoType": "",
    "rlsCFCacheValue": "A",
    "updateTime": "11:25",
    "vsamExtendedAddressing": false,
    "eattr": "",
    "forceSystemDeterminedBlockSize": false,
    "vsamRecordAccessBias": "",
    "compactionType": "",
    "blockSizeLimit": "",
    "dynamicVolumeCount": "",
    "dsnType": ""
  }
]
```

Figure 264. Sample response body

Get a data class definition

This operation retrieves a data class definition. The response content contains properties of the data class.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/dataclasses/<dc-name>
```

Query parameters

You can include query parameters on this request to specify the data class name and filter the level of detail in the returned content. For descriptions of the query parameters, see [Table 349 on page 598](#).

Table 349. Query parameters for retrieving a data class definition

Query Parameter	Description	Required or Optional	Type
dc-name	Specifies a data class name. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none"> Contains 1 - 8 alphanumeric characters Begins with an alphabetic or national character. National characters are \$, #, and @. Wildcards are not allowed. 	Required	String
detail-data	Specifies whether the response contains detailed information. Valid values are Y and N. A Y value requests detailed information. A N value requests summary information. The default value is N. (N is used if Y is not specified.)	Optional	Boolean

Description

On successful completion, the request returns the properties of a data class. The response body contains summary information or detailed information, according to the query parameters.

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

HTTP status codes

For a successful request, the response body contains an array of JSON data classes documents that match the specifications. If you do not request detailed information, each document contains only the information in [“Response content: Summary record”](#) on page 598. If you request detailed information, each document contains the information in [“Response content: Detailed record”](#) on page 611.

For a list of status codes, see [“Error handling”](#) on page 548.

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 350](#) on page 598.

Table 350. Response content for a successful retrieval of data classes summary information

Field name	Description
dataClassName	Data class name, consisting of 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
lastUser	User ID of the user that made the last update to the data class definition.
updateDate	Date that the storage class definition was last changed.
updateTime	Time that the storage class definition was last changed.
recorg	Specifies how VSAM data sets allocated by a data class are organized. Possible values include: ES VSAM Entry Sequenced Data Set KS VSAM Keyed Sequential Data Set LS VSAM Linear Space Data Set RR VSAM Relative Record Data Set Blank If the value has not been specified, the value will be blank.
recfm	Specifies the record format and type of carriage control assigned to non-VSAM data sets. Possible values include: A ANSI carriage control B Blocked format F Fixed format M Machine carriage control S Standard (if format is fixed) or Spanned (if format is variable) U Undefined format V Variable format Blank If the value has not been specified, the value will be blank.
vsamExtendedAddressing	Specifies whether extended addressability is provided. Extended addressability provides data sets with addressability of more than 4 gigabytes. Possible values include: TRUE Provides extended addressability. FALSE Does not provide extended addressability.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
spaceConstraintRelief	<p>Specifies whether space constraint relief method was selected or not during data class define/alter.</p> <p>Possible values include:</p> <p>TRUE Re-allocation methods are to be used by SMS in case of allocation failure.</p> <p>FALSE No attempt of retry are to be used by SMS in case of allocation failure.</p>
vsamKeyOffset	<p>Distance (in bytes) from the start of the record to the start of the key field. For non-VSAM data sets, KEYOFF is ignored.</p> <p>Possible values: 0 - 32760. If the value has not been specified, the value will be blank.</p>
vsamKeyLength	<p>Distance (in bytes) from the start of the record to the start of the key field. For non-VSAM data sets, KEYOFF is ignored.</p> <p>Possible values: 0 - 255 for non-VSAM data sets and 1 - 255 for key-sequenced VSAM data sets. If the value has not been specified, the value will be blank.</p>
primarySpaceAmount	<p>Amount of space the data class initially allocated for a data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>
secondarySpaceAmount	<p>Additional space that can be allocated for a data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>
directoryBlocks	<p>Number of blocks allocated for the directory of a partitioned data set.</p> <p>Possible values: 0 - 999999. If the value has not been specified, the value will be blank.</p>
avgrec	<p>Specifies whether the data class allocates space in bytes, kilobytes, or megabytes.</p> <p>Possible values include:</p> <p>K Space is allocated in kilobytes.</p> <p>M Space is allocated in megabytes.</p> <p>U Space is allocated in bytes.</p> <p>Blank If the value has not been specified, the value will be blank.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
reduceSpaceUpTo	Specifies the amount in % to be used by SMS for reducing the requested space quantity during data set allocation failure. Possible values: 0 - 99. If the value has not been specified, the value will be blank.
vsamRecordAccessBias	Allows the system to acquire and choose the buffering algorithms. Possible values include: SYSTEM Allows the system to choose the number of buffers and the buffering algorithms for the VSAM data set. USER Number of buffers and buffering algorithms will be based on user specified (or defaulted) values and current algorithms. DO Indicates system managed buffering with direct optimization. DW Indicates system managed buffering with sequential optimization. SO Indicates system managed buffering with sequential optimization. SW Indicates system managed buffering weighted for sequential processing. Blank If the value has not been specified, the value will be blank.
dynamicVolumeCount	Maximum number of volumes that DFSMS can dynamically add to an SMS managed data set. Possible values: 1 - 59. If the value has not been specified, the value will be blank.
avgValue	Multiplication factor used in determining allocated space. Possible values: 0 - 65535. If the value has not been specified, the value will be blank.
spaceOverrideFlag	Specifies whether the data class space attributes will override the space attributes from other sources, such as JCL. Possible values: TRUE Data class attributes will override others. FALSE Data class attributes can be overridden.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
forceSystemDeterminedBlockSize	<p>Specifies whether the system will ignore a user-specified block size if no program opens the data set for writing while it still is allocated. This prevents the user from overriding a system-determined block size.</p> <p>Possible values:</p> <p>TRUE If no program opens the data set for writing while the new data set still is allocated, then the system will discard a BLKSIZE value coded by the user. The system will attempt to determine an optimal block size. If a program opens the data set for output while it still is allocated, then the user-specified BLKSIZE will take effect and override a system-determined block size.</p> <p>FALSE If the user specifies a BLKSIZE value, it will take effect and override a system-determined block size. This is the normal way for the system to run.</p>
eattr	<p>Specifies whether the data set can support extended attributes DSCBs.</p> <p>Possible values:</p> <p>NO No extended attribute DSCBs. Only a format 1 DSCB will be created for this data set even if the data set is allocated on a volume that supports extended attribute DSCBs.</p> <p>OPT Extended attribute DSCBs are optional. Format 8 and 9 DSCBs will be created for this data set when allocated on a volume that supports extended attribute DSCBs. Otherwise, a format 1 DSCB will be created.</p> <p>Blank Not specified and defaults are used.</p> <p>For non-VSAM data sets, the EATTR value used by the system is equivalent to NO. For VSAM data sets, the EATTR value used by the system is equivalent to OPT.</p>
vsamSMBRMode31	<p>Specifies where the buffers and control blocks are to reside.</p> <p>Possible values:</p> <p>ALL Buffers and control blocks reside above the line.</p> <p>BUFF Buffers only reside above the line.</p> <p>CB Control blocks reside above the line.</p> <p>NONE Buffers and control blocks reside below the line.</p> <p>Blank If the value has not been specified, the value will be blank.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
recordLength	<p>Logical record length (in bytes) used when allocating data sets in this data class. The value displayed is the length of fixed-length records or the maximum length of variable-length records.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 1 - 32761 or blank for non-VSAM data sets • 1 - 32760 or blank for VSAM data sets. • If the value has not been specified, the value will be blank.
ciSize	<p>Number of bytes allocated for each control interval in the data portion, not the index portion, of a data set. To allow for overhead processing, the CISIZE value will be at least seven bytes greater than maximum record size.</p> <p>Possible values: 1 - 32768. If the value has not been specified, the value will be blank.</p>
ciFreeSpace	<p>Specifies what percentage of each control interval in a key-sequenced VSAM data set should be set aside as free space. VSAM uses the space to lengthen or insert records, as needed.</p> <p>Possible values: 0 - 100. If the value has not been specified, the value will be blank.</p>
caFreeSpace	<p>Specifies what percentage of each control area in a key-sequenced VSAM data set should be set aside as free space. VSAM uses the space to lengthen or insert records, as needed.</p> <p>Possible values: 0 - 100. If not specified, the value will be blank.</p>
vsamXSystemShareOptions	<p>Specifies how a VSAM data set can be shared among systems.</p> <p>Possible values:</p> <p>3</p> <p>All users can read and update the data set. VSAM does not protect the data set.</p> <p>4</p> <p>All users can read and update the data set. VSAM monitors access of data sets in order to prevent lost, damaged, or altered data.</p> <p>Blank</p> <p>If the value has not been specified, the value will be blank.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
vsamXRegionShareOptions	<p>Specifies how a VSAM data set can be shared among regions of one system or across regions of multiple systems.</p> <p>Possible values:</p> <p>1 All users can read the data set OR one user can update it.</p> <p>2 All users can read the data set AND one user can update it.</p> <p>3 All users can read and update the data set. VSAM does not protect the data set.</p> <p>4 All users can read and update the data set. VSAM helps prevent lost, damaged or altered data.</p> <p>Blank If the value has not been specified, the system will use the default value of primary.</p>
additionalVolumeAMT	<p>Type of allocation amount when a VSAM data set in extended format begins allocation on subsequent new volumes.</p> <p>Possible values:</p> <p>PRIMARY Primary allocation amount has been requested.</p> <p>SECONDARY Secondary allocation amount has been requested.</p> <p>Blank If the value has not been specified, the system will use the default value of primary.</p>
volumeCount	<p>Maximum number of volumes that can be used to store your data set.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • 1 - 59 (DASD) • 1 - 255 (Tape) • If the value has not been specified, the value will be blank.
dsnType	<p>Data set name type.</p> <p>Supported types include:</p> <ul style="list-style-type: none"> • PDS • LIBRARY • HFS • EXTENDED PREFERRED • EXTENDED REQUIRED • LARGE • If the value has not been specified, the value will be blank.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
retpdYear	<p>Default retention period or expiration date of data sets in this data class. Data sets are deleted or archived one day after the retention period or on the expiration date. Each data class can have a retention period or an expiration date (or neither), but not both.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • If retpdYear is blank, the expiration date is not specified. • If retpdAbsoluteDayOfYear is blank, the retention period is not specified. • If the retpdYear and retpdAbsoluteDayOfYear are not blank, the expiration date is specified. • If retpdYear is blank and retpdAbsoluteDayOfYear is not blank, retention period is specified. • Year range: 1900-2155.
retpdAbsoluteDayOfYear	<p>Absolute day of year. Data sets expire in the number of days displayed (0 - 93000).</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
compactionType	<p>This property shows whether tape volumes or DASD data sets associated with this data class are compacted or compressed. Compaction and compression increase storage capacity.</p> <p>Possible values:</p> <p>YES Extended format data sets are compressed and tape volumes are compacted. The type of DASD compression depends on the COMPRESS option in IGDSMSxx. Tape volumes are compacted unless overridden by the user through JCL/dynamic allocation.</p> <p>NO Data sets are not compressed. Tape volumes are not compacted, unless requested by the user on JCL/dynamic allocation.</p> <p>TCOM Extended format data sets are compressed using tailored dictionaries overriding SYS1.PARMLIB.</p> <p>GEN Extended format data sets are compressed using generic dictionaries overriding SYS1.PARMLIB.</p> <p>ZR The system will fail the allocation request if the zEDC function is not supported by the system or the minimum allocation amount requirement is not met.</p> <p>ZP The system will not fail the allocation request but rather create either a tailored compressed data set if the zEDC function is not supported by the system or create a non-compressed extended format data set if the minimum allocation amount requirement is not met.</p> <p>Blank Data sets are not compressed. Tape volumes may be compacted depending on what was specified by the user on JCL/dynamic allocation, the installation with the COMPACT option in parmlib member DEVSUPxx, or the allocated hardware model.</p> <p>-- Unexpected error.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
mediaType	Media type. Possible values: <ul style="list-style-type: none"> • MEDIA1 - Cartridge system • MEDIA2 - Enhanced capacity cartridge system tape • MEDIA3 - ½ inch / 320 meter particle media (3590) • MEDIA4 - Double length (3590) • MEDIA5 - 3592 module J, 3592 Model E05 • MEDIA5 - 3592 module JW, 3592 Model E05 • MEDIA5 - 3592 module JV, 3592 Model E05 • MEDIA5 - 3592 module JR, 3592 Model E05 • MEDIA9 - 3592 model E05 • MEDIA10 - 3592 model E05 • If the value has not been specified, the value will be blank.
recordingTechnology	Recording technology type. Possible values: <ul style="list-style-type: none"> • 18 track • 36 track • 128 track • 256 track • 384 track • EFMT1 recording technology • EFMT2 recording technology • EEFMT2 recording technology • EFMT3 recording technology • EEFMT3 recording technology • EFMT4 recording technology • EEFMT4 recording technology • Blank, if not specified
vsamExtentionReuse	Specifies whether you can open the cluster repeatedly as a new cluster. Possible values: TRUE Cluster is reusable. FALSE Cluster is not reusable.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
initialLoad	<p>Specifies whether the storage allocated to the data component was preformatted before records were inserted during initial load.</p> <p>Possible values:</p> <p>SPEED Indicates the data component's space was not preformatted.</p> <p>RECOVERY Indicates the data component's space was preformatted.</p>
blockSizeLimit	<p>Specifies the largest value that the system can determine for a data set block size when a program opens a dataset for output and no value is available. The system determines a data set block size that is appropriate for the media type when the data set is sequential, the record format is fixed or variable, and the logical record length is known. The BLKSZLIM keyword on the DD statement overrides this value. Limit the block size when a system or application that may read the tape does not support large blocks. Large blocks generally are more efficient.</p> <p>The actual block size determined by the system may be less than the value displayed.</p> <p>Possible values: 32760 - 2147483648. If the value has not been specified, the value will be blank.</p>
extentConstraintRemoval	<p>Specifies whether a VSAM data set is allowed to go beyond the 255 extents limit.</p> <p>Possible values:</p> <p>TRUE Remove the 255 extents limit.</p> <p>FALSE Keep the limit of 255 extents.</p>
logReplicate	<p>Specifies whether the data set will be eligible for replication.</p> <p>Possible values:</p> <p>TRUE Data set is eligible for VSAM replication.</p> <p>FALSE Data set is not eligible for VSAM replication.</p>
gsSpaceReduction	<p>Specifies whether space reduction on guaranteed space allocations is permitted or not.</p> <p>Possible values:</p> <p>TRUE Space reduction on guaranteed space allocations is permitted.</p> <p>FALSE Space reduction on guaranteed space allocations is not permitted.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
bwoType	<p>Backup While Open (BWO) type.</p> <p>Possible values:</p> <p>TYPECICS BWO processing for CICS VSAM file control data sets is permitted.</p> <p>TYPEIMS BWO processing for IMS VSAM data sets is to be used.</p> <p>NO BWO does not apply to the cluster.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
recordSpansCIAbility	<p>Specifies whether the data record is allowed to cross control interval boundaries.</p> <p>Possible values:</p> <p>SPANNED Record is contained in more than one control interval.</p> <p>NONSPANNED Record is contained in one control interval.</p> <p>Blank If the value has not been specified, the value will be blank.</p>
frlog	<p>Specifies whether to do the logging for changes to RCC forward recovery log stream.</p> <p>Possible values:</p> <p>ALL Tells VSAM to log both forward and backward recovery logging.</p> <p>NONE No logging in effect.</p> <p>REDO Tells VSAM to do forward recovery logging.</p> <p>UNDO Tells VSAM to do backward recovery logging.</p> <p>Blank Attribute not specified.</p>
logStreamID	Specifies the name of the recovery logstream.

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
rlsCFCacheValue	<p>Specifies that VSAM RLS data with greater than 4K CI's defined to DFSMS CF cache structures can be cached.</p> <p>Possible values:</p> <p>A - ALL Indicates that all of the data is cached for the sphere.</p> <p>N - NONE Indicates that none of the data will be cached.</p> <p>U - UPDATESONLY Indicates that only updated CI's will be cached.</p> <p>D - DIRONLY Indicates that the directory only will be cached.</p>
rlsAbove2GBBar	<p>Specifies whether virtual storage for RLS can be above the 2 Gigabyte bar.</p> <p>Possible values:</p> <p>TRUE Place buffers above the bar in the SMSVSAM address space.</p> <p>FALSE Do not place any data in storage located above the bar.</p>
caReclaim	<p>Specifies whether the DASD space for empty CAs may be reused on z/OS V1.12 or later systems.</p> <p>Possible values:</p> <p>TRUE Free CAs are reused.</p> <p>FALSE Free CAs are not reused.</p>
scalingOption	<p>Tape Support use.</p> <p>Possible values:</p> <p>TRUE Scale to optimal performance</p> <p>FALSE To use the full capacity (no scaling).</p>
segmentingOption	<p>Tape Support use.</p> <p>Possible values:</p> <p>TRUE Enable segmentation format.</p> <p>FALSE No segmentation.</p>
unitForSystemManagedBufferValue	<p>Specifies unit in KB or MB for VSP value.</p>

Table 350. Response content for a successful retrieval of data classes summary information (continued)

Field name	Description
systemManagedBuffer	Amount of virtual storage for SMB Direct Access Bias obtained for buffers when opening the data set. Possible values: <ul style="list-style-type: none"> • 1KB - 2048000KB • IMB - 2048MB
dataSetKeyLabelName	Specifies the label for the encryption key used by the access methods. Possible values: 1 - 52 characters, typically containing alphanumeric, national or special characters with some additional characters also being allowed. If DASD DATA SET KEY LABEL length is more than 52 characters, the first 49 characters followed by '...' are displayed. The rest of DASD DATA SET KEY LABEL is truncated.

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 350](#) on page 598 and [Table 351](#) on page 611.

Table 351. Response content for a successful retrieval of data classes detailed information

Field name	Description
dataClassNameLength	Specifies data class name length.
description	Specifies description of data class.
blockedOrUnblocked	AMS use only. Specifies blocked or unblocked. Possible values: <ul style="list-style-type: none"> • TRUE (blocked) • FALSE (unblocked/null)
standardOrSpanned	AMS use only. Specifies standard or spanned. Possible values: <ul style="list-style-type: none"> • TRUE (standard) • FALSE (spanned)
indexOptionImbed	Specifies VSAM data set properties index option - imbed Possible values: True (IMBED) or False (No)
indexOptionReplicate	Specifies VSAM data set properties index option - replicate Possible values: TRUE (replicate) and FALSE (do not replicate)
overrideJCLSpecified	Indicates whether override JCL is specified. Possible values: True or False

Table 351. Response content for a successful retrieval of data classes detailed information (continued)

Field name	Description
sphereRecoverability	Indicates whether sphereRecoverability is specified. Possible values: 0 Not specified 1 Non-recoverable sphere 2 UNDO 3 ALL 4 REDO
logStreamIDLength	Indicates logStreamID length.
dataSetKeyLabelLength	Indicates data set key label length.
keyLabel1Length	Indicates key label 1 length.
keyLabel1Name	Indicates key label 1 name.
keyCode1	Indicates key code 1.
keyLabel2Length	Indicates key label 2 length.
keyLabel2Name	Indicates key label 2 name.
keyCode2	Indicates key code 2.

Example HTTP interaction

1. The following example shows a request to retrieve a list of all data classes on the system with detailed information about each data class:

```
GET /zosmf/storage/rest/v1/dataclasses/<dc-name>
```

Figure 265. Sample request to get a list of data classes with detailed information about each data class

The following is the response body for the request.

```
[
  {
    "lastUser": "MCKENZI",
    "dataSetKeyLabelLength": 0,
    "recordLength": "",
    "caFreeSpace": "",
    "logStreamID": "",
    "scalingOption": false,
    "retpdAbsoluteDayOfYear": "",
    "spaceOverrideFlag": false,
    "directoryBlocks": "",
    "spaceConstraintRelief": false,
    "recorg": "",
    "secondarySpaceAmount": "",
    "extentConstraintRemoval": false,
    "keyCode1": "",
    "systemManagedBuffer": "",
    "keyCode2": "",
    "unitForSystemManagedBufferValue": "MB",
    "additionalVolumeAMT": "",
    "logReplicate": false,
    "avgValue": "",
    "gsSpaceReduction": false,
    "recfm": "",
    "vsamXRegionShareOptions": "",
    "mediaType": "",
    "frlog": "",
    "vsamXSystemShareOptions": "",
    "overrideJCLSpecified": false,
    "keyLabel1Length": 0,
    "caReclaim": true,
    "vsamExtentionReuse": false,
    "ciFreeSpace": "",
    "vsamKeyLength": "",
    "indexOptionImbed": false,
    "dataSetKeyLabelName": "",
    "reduceSpaceUpTo": "",
    "updateDate": "2005/01/08",
    "sphereRecoverability": 0,
    "rlsAbove2GBBar": false,
    "segmentingOption": false,
    "description": "SMS-MANAGED EXTENDED FORMAT SEQUENTIAL DATA SET FOR STAND-ALONE DUMP",
    "initialLoad": "RECOVERY",
    "avgrec": "",
    "keyLabel2Length": 0,
    "keyLabel2Name": "",
    "recordSpansCIAbility": "",
    "logStreamIDLength": 0,
    "primarySpaceAmount": "",
    "dataClassName": "EFDUMPDS",
    "recordingTechnology": "",
    "vsamKeyOffset": "",
    "ciSize": "",
    "volumeCount": 1,
    "blockedOrUnblocked": false,
    "vsamSMBRMode31": "",
    "retpdYear": "",
    "bwoType": "",
    "rlsCFCacheValue": "A",
    "indexOptionReplicate": false,
    "dataClassNameLength": 8,
    "updateTime": "19:33",
    "vsamExtendedAddressing": true,
    "eattr": "",
    "standardOrSpanned": false,
    "forceSystemDeterminedBlockSize": false,
    "vsamRecordAccessBias": "USER",
    "compactionType": "",
    "blockSizeLimit": "",
    "dynamicVolumeCount": "",
    "dsnType": "EXTENDED REQUIRED",
    "keyLabel1Name": ""
  }
]
```

Figure 266. Sample response body

Get a list of storage classes

This operation retrieves a list of storage class records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all storage classes that match the query parameters.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/storageclasses
```

Query parameters

You can include query parameters on this request to filter storage classes and determine the level of detail in the returned content. For descriptions of the query parameters, see [Table 352 on page 614](#).

Table 352. Query parameters for retrieving a list of storage classes

Query Parameter	Description	Required or Optional	Type
filter	Specifies a storage class name. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none">• Contains 1-8 alphanumeric characters.• Begins with an alphabetic or national character. National characters are \$, #, and @.• The asterisk (*) and percent (%) wildcards are allowed.<ul style="list-style-type: none">– Asterisk matches any number of characters.– Percent matches one character.– Asterisk and percent wildcards can be used together.– You can use more than one asterisk and more than one percent wildcard.	Optional	String
detail-data	Specifies whether the response contains detailed information. Valid values are Y and N. A Y value requests detailed information. A N value requests summary information. The default value is N. (N is used if Y is not specified.)	Optional	String

Table 352. Query parameters for retrieving a list of storage classes (continued)

Query Parameter	Description	Required or Optional	Type
offset	<p>Specifies the offset for the list of storage classes. The request retrieves storage classes starting from this offset value. For example, an offset of 50 means retrieve from the 51st position in the list and onward. The valid range of offset values is 0 - 2147483647.</p> <p>The offset cannot be larger than or equal to the number of storage classes.</p> <p>The offset cannot be used with the filter parameter in the same request.</p> <p>The default value for offset is 0.</p>	Optional	Number
limit	<p>Specifies the maximum number of storage classes that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>You can use the limit parameter with the offset parameter. For example, if you specify offset = 1 and limit = 3, the request returns the second, third, and fourth storage classes from the result set.</p> <p>Observe the following usage notes:</p> <ul style="list-style-type: none"> • If you specify limit = 0, the request returns all of the queried storage classes from the offset value. • If the offset plus limit is a larger value than the number of queried storage classes, the request returns only the number of the queried storage classes, minus the offset. • If the offset plus limit is a smaller value than the number of queried storage classes, the request returns only the limit number of storage classes, starting from the offset. <p>The limit cannot be used with the filter parameter in the same request.</p> <p>The default value for limit is 0.</p>	Optional	Number

Description

On successful completion, the request returns a list of storage class records. The records contain summary information or detailed information, according to the query parameters. The returned content includes records for all storage classes that match the query parameters. For descriptions of the query parameters that you can specify on the request, see [Table 352 on page 614](#).

For descriptions of the response content, see [“Response content: Summary record” on page 616](#) and [“Response content: Detailed record” on page 619](#).

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

HTTP status codes

For a successful request, the response body contains an array of JSON storage classes documents that match the specifications. If you do not request detailed information, each document contains only the information in [“Response content: Summary record” on page 616](#). If you request detailed information, each document contains the information in [“Response content: Detailed record” on page 619](#).

For a list of status codes, see [“Error handling” on page 548](#).

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 353 on page 616](#).

Table 353. Response content for a successful retrieval of storage classes summary information	
Field name	Description
storageClassName	Storage class name, consisting of 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.
lastUser	User ID of the user that made the last update to the storage class definition.
updateDate	Date that the storage class definition was last changed.
updateTime	Time that the storage class definition was last changed.
graranttedSpace	<p>Specifies whether you can preallocate space for multi-volume data sets. If the field shows true, multi-volume data sets can be preallocated with the same or different amounts of space on more than one volume. If the field shows false, the Storage Management Subsystem will not allow preallocation of multi-volume data sets.</p> <p>Possible Value: True or false</p>
syncDev	<p>Specifies whether the system should return from a BSAM (or WAIT) issued for a WRITE against a PDSE member or a compressed format data set before (resynchronized) or after (synchronized) the data set has actually been written to a storage device.</p> <p>Possible values:</p> <p>TRUE Indicates synchronized write.</p> <p>FALSE Indicates no synchronization.</p>
directMillisecondResponseTime	<p>Number of milliseconds required to read or write a 4-kilobyte block of data.</p> <p>Possible values: 1 - 999. If the value is not specified, the value will be blank.</p>

Table 353. Response content for a successful retrieval of storage classes summary information (continued)

Field name	Description
sequentialMillisecondResponseTime	<p>Number of milliseconds required to read or write a 4-kilobyte block of data. Three decimal places have been reserved to support future sub-millisecond response times.</p> <p>Possible values: 1 - 999. If the value is not specified, the value will be blank.</p>
directBias	<p>Direct access bias for data sets in the storage class. Specifies whether the majority of I/O scheduled for the data sets in the storage class is READ, WRITE or unknown. Transaction logs usually have a WRITE bias. A rarely updated production PROCLIB would have a READ bias.</p> <p>Possible values: R (READ), W (WRITE) or blank</p>
seqBias	<p>Sequential access bias for data sets in the storage class. Specifies whether the majority of I/O scheduled for the data sets in the storage class is READ, WRITE or unknown. Transaction logs usually have a WRITE bias. A rarely updated production PROCLIB would have a READ bias.</p> <p>Possible values: R (READ), W (WRITE) or blank</p>
availabilityOptions	<p>Specifies whether data set processing will continue after device failures.</p> <p>Possible values:</p> <p>CONTINUOUS Processing of a data set continues if a device failure severs communications with the volume that contains the data set. Data is placed on a dual copy, a RAMAC Array Sub- system, or a RAMAC Array DASD volume.</p> <p>STANDARD Continuous processing is unavailable after device failures. Data is placed on a non-dual copy or RAID volume. Non-RAID is preferred to RAID (RAMAC) volume.</p> <p>PREFERRED Data may be placed on devices that support continuous processing. RAID is preferred to non-RAID.</p> <p>NOPREF Data is placed on any volume. There is no preference among volumes. Processing availability depends on the type of the volume in which the data is placed.</p> <p>Note: This option does not apply to dual-copy volumes.</p>

Table 353. Response content for a successful retrieval of storage classes summary information
(continued)

Field name	Description
accessibility	<p>Specifies whether the data sets in the storage class should be allocated to volumes supported by Concurrent Copy. When used with the ABACKUP/BACKUP COPY TECHNIQUE attributes of the management class, this field determines if the data sets should retain continuous write access during backup.</p> <p>Possible values:</p> <p>CONTINUOUS Data sets must be allocated to volumes supported by Concurrent Copy. The allocation fails for data sets that cannot be allocated to such volumes.</p> <p>CONT PREFD Data sets should be allocated to volumes supported by Concurrent Copy. If this cannot be done, a data set may be allocated to a volume not supported by Concurrent Copy.</p> <p>STANDARD Data sets should be allocated to volumes not supported by Concurrent Copy. If this cannot be done, a dataset may be allocated to a volume supported by Concurrent Copy.</p> <p>NOPREF Data sets should be allocated to volumes whether the volumes support Concurrent Copy or not.</p>
initialAccessResponse	<p>Time required (in seconds) to locate, mount, and prepare media for data transfer.</p> <p>Possible values: 0 - 9999. If the value is not specified, the value will be blank.</p>
stripingSustainedDataRate	<p>Sequential data transfer rate desired for striped data sets in this storage class. The system uses this value to determine the number of stripes it will attempt to allocate for the data sets.</p> <p>Possible values: 0 - 9999. If the value is not specified, the value will be blank.</p>
cacheSetName	<p>Name of the CF cache set that is defined in the CDS for the specified storage class.</p>
directCFWeight	<p>Specifies the relative importance of data in a storage class when it is accessed directly. Storage classes with higher CF weight values tend to have greater cache resources allocated than storage classes with lower weight values.</p> <p>Possible values: 1 - 11 numeric value, 1 - 2 characters. If the value is not specified, the value will be blank.</p>
sequentialCFWeight	<p>Specifies the relative importance of data in a storage class when it is accessed sequentially. Storage classes with higher CF weight value tend to have more CF cache resources allocated than storage classes with lower weight values.</p> <p>Possible values: 1 - 11 numeric value, 1 - 2 characters. If the value is not specified, the value will be blank.</p>

Table 353. Response content for a successful retrieval of storage classes summary information (continued)

Field name	Description
multitieredValueValidOnly	<p>Enables SMS to attempt to allocate using a volume in the first listed storage group prior to allocating in subsequent storage groups.</p> <p>Possible values:</p> <p>TRUE SMS will direct allocations to volumes which are enabled, and below threshold in the first storage group listed, followed by subsequent storage groups until the allocation is successful.</p> <p>FALSE No sequence order followed.</p>
disconnectSphereAtClose Value	<p>Specifies whether the sphere should be disconnected upon closing the data set or stay connected for a period of time.</p> <p>Possible values:</p> <p>TRUE Sphere will be disconnected upon closing.</p> <p>FALSE Sphere stays connected for a period of time.</p>
pavOptionsValue	<p>Parallel Access Volume Capability specifies volume selection algorithms.</p> <p>REQUIRED PAV Capability is required. Only those volumes that support this capability will be eligible. All other volumes will be rejected from consideration.</p> <p>PREFERRED Capability is preferred. Volumes with this capability will be preferred over volumes that do not have this capability.</p> <p>STANDARD Volumes without PAV capability will be preferred over volumes that have this capability.</p> <p>NOPREF Volumes with or without PAV capability will be equally considered for volume selection. This is the default value.</p>
oslValue	<p>Specifies the sublevel of an OAM disk or tape hierarchy level to use for object storage. The OAM SUBLEVEL is only applicable when the INITIAL ACCESS RESPONSE SECONDS value is equal to 0, or when the INITIAL ACCESS RESPONSE SECONDS value is greater than 0 and the SUSTAINED DATA RATE value is greater than or equal to 3.</p> <p>Possible values: 1 - 2. If the value is not specified, the value will be blank.</p>
lockSetName	Name of the CF lock set associated with the storage class.

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 353 on page 616](#) and [Table 354 on page 620](#).

Table 354. Response content for a successful retrieval of storage classes detailed information

Field name	Description	
storageClassNameLength	Specifies storage class name length.	
description	Specifies description of storage class.	
dataSetSeparation	Specifies data set separation. Possible values: True or False	
accVersionParm	Possible values: 0 Blank 1 Yes 2 No	
accBackupParm	Possible values: 0 Blank 1 Yes 2 No	
cacheSetNameLength	Specifies cache set name length.	
lockSetNameLength	Specifies lock set name length.	

Example HTTP interaction

1. The following example shows a request to retrieve a list of all storage classes on the system with summary information about each storage class:

```
GET /zosmf/storage/rest/v1/storageclasses
```

Figure 267. Sample request to get a list of storage classes with summary information about each storage class

The following is the response body for the request.

```

{
  "cacheSetName": "",
  "multitieredValueValidOnly": false,
  "lastUser": "KNMORT",
  "updateDate": "2003/06/30",
  "disconnectSphereAtCloseValue": false,
  "pavOptionsValue": "NOPREF",
  "accessibility": "NOPREF",
  "seqBias": "W",
  "sequentialCFWeight": "",
  "updateTime": "14:44",
  "lockSetName": "",
  "directBias": "W",
  "oslValue": "",
  "directMillisecondResponseTime": 5,
  "initialAccessResponse": "",
  "graranttedSpace": true,
  "stripingSustainedDataRate": "",
  "sequentialMillisecondResponseTime": 5,
  "storageClassName": "ARCL0G",
  "syncDev": false,
  "directCFWeight": "",
  "availabilityOptions": "NOPREF"
},
{
  "cacheSetName": "",
  "multitieredValueValidOnly": false,
  "lastUser": "KNMORT",
  "updateDate": "2003/06/30",
  "disconnectSphereAtCloseValue": false,
  "pavOptionsValue": "NOPREF",
  "accessibility": "NOPREF",
  "seqBias": "",
  "sequentialCFWeight": "",
  "updateTime": "14:43",
  "lockSetName": "",
  "directBias": "",
  "oslValue": "",
  "directMillisecondResponseTime": "",
  "initialAccessResponse": "",
  "graranttedSpace": false,
  "stripingSustainedDataRate": "",
  "sequentialMillisecondResponseTime": "",
  "storageClassName": "ATL",
  "syncDev": false,
  "directCFWeight": "",
  "availabilityOptions": "NOPREF"
}
]

```

Figure 268. Sample response body

Get a storage class definition

This operation retrieves a storage class definition. The response content contains properties of the storage class.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/storageclasses/<sc-name>
```

Query parameters

You can include query parameters on this request to specify the storage class name and filter the level of detail in the returned content. For descriptions of the query parameters, see [Table 355 on page 622](#).

Table 355. Query parameters for retrieving a storage class definition

Query Parameter	Description	Required or Optional	Type
dc-name	Specifies a storage class name. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none"> Contains 1 - 8 alphanumeric characters Begins with an alphabetic or national character. National characters are \$, #, and @. Wildcards are not allowed. 	Required	String
detail-data	Specifies whether the response contains detailed information. Valid values are Y and N. A Y value requests detailed information. A N value requests summary information. The default value is N. (N is used if Y is not specified.)	Optional	Boolean

Description

On successful completion, the request returns the properties of a storage class. The response body contains summary information or detailed information, according to the query parameters.

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

HTTP status codes

For a successful request, the response body contains an array of JSON storage classes documents that match the specifications. If you do not request detailed information, each document contains only the information in [“Response content: Summary record”](#) on page 622. If you request detailed information, each document contains the information in [“Response content: Detailed record”](#) on page 626.

For a list of status codes, see [“Error handling”](#) on page 548.

Response content: Summary record

Successful queries with no request for details return information in fields that are described in [Table 356](#) on page 622.

Table 356. Response content for a successful retrieval of storage classes summary information

Field name	Description
storageClassName	Storage class name, consisting of 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.

Table 356. Response content for a successful retrieval of storage classes summary information (continued)

Field name	Description
lastUser	User ID of the user that made the last update to the storage class definition.
updateDate	Date that the storage class definition was last changed.
updateTime	Time that the storage class definition was last changed.
graranttedSpace	Specifies whether you can preallocate space for multi-volume data sets. If the field shows true, multi-volume data sets can be preallocated with the same or different amounts of space on more than one volume. If the field shows false, the Storage Management Subsystem will not allow preallocation of multi-volume data sets. Possible Value: True or false
syncDev	Specifies whether the system should return from a BSAM (or WAIT) issued for a WRITE against a PDSE member or a compressed format data set before (resynchronized) or after (synchronized) the data set has actually been written to a storage device. Possible values: TRUE Indicates synchronized write. FALSE Indicates no synchronization.
directMillisecondResponseTime	Number of milliseconds required to read or write a 4-kilobyte block of data. Possible values: 1 - 999. If the value is not specified, the value will be blank.
sequentialMillisecondResponseTime	Number of milliseconds required to read or write a 4-kilobyte block of data. Three decimal places have been reserved to support future sub-millisecond response times. Possible values: 1 - 999. If the value is not specified, the value will be blank.
directBias	Direct access bias for data sets in the storage class. Specifies whether the majority of I/O scheduled for the data sets in the storage class is READ, WRITE or unknown. Transaction logs usually have a WRITE bias. A rarely updated production PROCLIB would have a READ bias. Possible values: R (READ), W (WRITE) or blank
seqBias	Sequential access bias for data sets in the storage class. Specifies whether the majority of I/O scheduled for the data sets in the storage class is READ, WRITE or unknown. Transaction logs usually have a WRITE bias. A rarely updated production PROCLIB would have a READ bias. Possible values: R (READ), W (WRITE) or blank

Table 356. Response content for a successful retrieval of storage classes summary information
(continued)

Field name	Description
availabilityOptions	<p>Specifies whether data set processing will continue after device failures.</p> <p>Possible values:</p> <p>CONTINUOUS Processing of a data set continues if a device failure severs communications with the volume that contains the data set. Data is placed on a dual copy, a RAMAC Array Sub- system, or a RAMAC Array DASD volume.</p> <p>STANDARD Continuous processing is unavailable after device failures. Data is placed on a non-dual copy or RAID volume. Non-RAID is preferred to RAID (RAMAC) volume.</p> <p>PREFERRED Data may be placed on devices that support continuous processing. RAID is preferred to non-RAID.</p> <p>NOPREF Data is placed on any volume. There is no preference among volumes. Processing availability depends on the type of the volume in which the data is placed.</p> <p>Note: This option does not apply to dual-copy volumes.</p>
accessibility	<p>Specifies whether the data sets in the storage class should be allocated to volumes supported by Concurrent Copy. When used with the ABACKUP/BACKUP COPY TECHNIQUE attributes of the management class, this field determines if the data sets should retain continuous write access during backup.</p> <p>Possible values:</p> <p>CONTINUOUS Data sets must be allocated to volumes supported by Concurrent Copy. The allocation fails for data sets that cannot be allocated to such volumes.</p> <p>CONT PREFD Data sets should be allocated to volumes supported by Concurrent Copy. If this cannot be done, a data set may be allocated to a volume not supported by Concurrent Copy.</p> <p>STANDARD Data sets should be allocated to volumes not supported by Concurrent Copy. If this cannot be done, a dataset may be allocated to a volume supported by Concurrent Copy.</p> <p>NOPREF Data sets should be allocated to volumes whether the volumes support Concurrent Copy or not.</p>
initialAccessResponse	<p>Time required (in seconds) to locate, mount, and prepare media for data transfer.</p> <p>Possible values: 0 - 9999. If the value is not specified, the value will be blank.</p>

Table 356. Response content for a successful retrieval of storage classes summary information (continued)

Field name	Description
stripingSustainedDataRate	Sequential data transfer rate desired for striped data sets in this storage class. The system uses this value to determine the number of stripes it will attempt to allocate for the data sets. Possible values: 0 - 9999. If the value is not specified, the value will be blank.
cacheSetName	Name of the CF cache set that is defined in the CDS for the specified storage class.
directCFWeight	Specifies the relative importance of data in a storage class when it is accessed directly. Storage classes with higher CF weight values tend to have greater cache resources allocated than storage classes with lower weight values. Possible values: 1 - 11 numeric value, 1 - 2 characters. If the value is not specified, the value will be blank.
sequentialCFWeight	Specifies the relative importance of data in a storage class when it is accessed sequentially. Storage classes with higher CF weight value tend to have more CF cache resources allocated than storage classes with lower weight values. Possible values: 1 - 11 numeric value, 1 - 2 characters. If the value is not specified, the value will be blank.
multitieredValueValidOnly	Enables SMS to attempt to allocate using a volume in the first listed storage group prior to allocating in subsequent storage groups. Possible values: TRUE SMS will direct allocations to volumes which are enabled, and below threshold in the first storage group listed, followed by subsequent storage groups until the allocation is successful. FALSE No sequence order followed.
disconnectSphereAtCloseValue	Specifies whether the sphere should be disconnected upon closing the data set or stay connected for a period of time. Possible values: TRUE Sphere will be disconnected upon closing. FALSE Sphere stays connected for a period of time.

Table 356. Response content for a successful retrieval of storage classes summary information
(continued)

Field name	Description
pavOptionsValue	<p>Parallel Access Volume Capability specifies volume selection algorithms.</p> <p>REQUIRED PAV Capability is required. Only those volumes that support this capability will be eligible. All other volumes will be rejected from consideration.</p> <p>PREFERRED Capability is preferred. Volumes with this capability will be preferred over volumes that do not have this capability.</p> <p>STANDARD Volumes without PAV capability will be preferred over volumes that have this capability.</p> <p>NOPREF Volumes with or without PAV capability will be equally considered for volume selection. This is the default value.</p>
oslValue	<p>Specifies the sublevel of an OAM disk or tape hierarchy level to use for object storage. The OAM SUBLEVEL is only applicable when the INITIAL ACCESS RESPONSE SECONDS value is equal to 0, or when the INITIAL ACCESS RESPONSE SECONDS value is greater than 0 and the SUSTAINED DATA RATE value is greater than or equal to 3.</p> <p>Possible values: 1 - 2. If the value is not specified, the value will be blank.</p>
lockSetName	Name of the CF lock set associated with the storage class.

Response content: Detailed record

Successful queries for a detailed record return information in fields that are described in [Table 356 on page 622](#) and [Table 357 on page 626](#).

Table 357. Response content for a successful retrieval of storage classes detailed information

Field name	Description	
storageClassNameLength	Specifies storage class name length.	
description	Specifies description of storage class.	
dataSetSeparation	<p>Specifies data set separation.</p> <p>Possible values: True or False</p>	
accVersionParm	<p>Possible values:</p> <p>0 Blank</p> <p>1 Yes</p> <p>2 No</p>	

Table 357. Response content for a successful retrieval of storage classes detailed information (continued)

Field name	Description	
accBackupParm	Possible values: 0 Blank 1 Yes 2 No	
cacheSetNameLength	Specifies cache set name length.	
lockSetNameLength	Specifies lock set name length.	

Example HTTP interaction

1. The following example shows a request to retrieve a list of all storage classes on the system with detailed information about each storage class:

```
GET /zosmf/storage/rest/v1/storageclasses/<sc-name>
```

Figure 269. Sample request to get a list of storage classes with detailed information about each storage class

The following is the response body for the request.

```

[
  {
    "cacheSetName": "",
    "accVersionParm": "",
    "lastUser": "KNMORT",
    "updateDate": "2003/06/30",
    "accBackupParm": "",
    "disconnectSphereAtCloseValue": false,
    "directCFWeightSpecified": false,
    "accessibility": "NOPREF",
    "availabilityFlag": true,
    "seqResponseTimeObjective": true,
    "description": "",
    "lockSetName": "",
    "cacheSetNameLength": 0,
    "directBias": "W",
    "multilockSetSpecified": false,
    "initialAccessResponse": 0,
    "dataSetSeparation": false,
    "storageClassNameLength": 6,
    "disconnectSphereAtClose": false,
    "syncDev": false,
    "directBiasFlag": true,
    "directCFWeight": 0,
    "availabilityOptions": "NOPREF",
    "multitieredValueValidOnly": false,
    "ACCVersionigSpecified": false,
    "pavSpecified": false,
    "pavOptionsValue": "NOPREF",
    "multiptieredSpecified": false,
    "seqBias": "W",
    "seqBiasFlag": true,
    "sequentialCFWeight": false,
    "ACCBakupSpecified": false,
    "updateTime": "14:44",
    "olsSpecified": false,
    "directMillisecondResponseTime": 5,
    "directResponseTimeObjective": true,
    "initialAccessResponseFlag": false,
    "olsValue": 0,
    "graranttedSpace": true,
    "stripingSustainedDataRate": false,
    "sequentialMillisecondResponseTime": 5,
    "accessibilityFlag": true,
    "storageClassName": "ARCL0G",
    "lockSetNameLength": 0
  }
]

```

Figure 270. Sample response body

Define a volume list to a storage group

Use this operation to add a volume list to a storage group.

HTTP method and URI path

```
POST /zosmf/storage/rest/<version>/storagegroups/<stg-name>/volumes
```

Query parameters

Table 358. Query parameters for retrieving a list of volumes

Query Parameter	Description	Required or Optional	Type
stg-name	Specifies the name of a storage-group that is associated with the volume. The value is not case-sensitive. The value must meet the following criteria: <ul style="list-style-type: none">• Contains 1-8 alphanumeric characters• Begins with an alphabetic or national character. National characters are \$, #, and @.• Wildcards are not allowed.• Only storage groups with POOL, DUMMY, or COPY POOL BACKUP types can be added volumes.	Required	String
validate	Specifies whether to validate the SCDS after adding volume list to a storage group Valid values are Y and N.Y value requests to validate the SCDS. N value requests not to validate SCDS. The default value is N if the parameter is not specified.	Optional	String
activate	Specifies whether to activate SCDS after adding volume list to a storage group. Valid values are Y and N. Y value requests to activate SCDS. N value requests not to activate SCDS. If <i>activate</i> is Y, the REST API will always validate the SCDS before activation, regardless of the value defined for <i>validate</i> . The default value is N if the parameter is not specified.	Optional	String

Description

This operation adds one or more new volume serial numbers to the volume list of a storage group in an SMS source control data set. It is recommended to backup the SCDS before invoking this REST API so that you are able to restore it if any error happens when adding volumes, validation or activation.

On successful completion, the request returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 returned with an object containing new volumes indicates success.

Request content

The request content is expected to contain a JSON object. See [Table 359 on page 630](#).

Table 359. Request content for the define volume list for storage group request

Field name	Description	Required or Optional	Type
SCDS	Specifies the SMS source control data set. The SCDS name can be up to 42 characters, including delimiters (periods). This property is not case-sensitive.	Required	String
range	Specifies the array which is used to define volumes. Each element can contain values for prefix, from, to, suffix and type. This property is not case-sensitive .	Required	Array
prefix	Specifies either a volume serial number or the first common characters in a range of volume serial numbers. If you enter part of a range, enter the remaining distinctive characters of the volume serial numbers in the FROM and TO fields and the SUFFIX field where applicable. Enter up to six characters. This property is not case-sensitive.	Optional	String
from	Specifies the characters unique to the first volume serial number in a range. Use the PREFIX and SUFFIX fields to specify the leading and trailing characters. Valid entries are decimal or hexadecimal numbers or alphabetic. This property is not case-sensitive.	Optional	String
to	Specifies the characters unique to the last volume serial number in a range. Use the PREFIX and SUFFIX fields to specify the leading and trailing characters. Valid entries are decimal or hexadecimal numbers or alphabetic. This property is not case-sensitive.	Optional	String
suffix	Specifies the common trailing characters in a range of volume serial numbers. You can choose to use this if you specify values in the FROM and TO fields. Enter up to five characters. This property is not case-sensitive.	Optional	String

Table 359. Request content for the define volume list for storage group request (continued)

Field name	Description	Required or Optional	Type
type	When you specify hexadecimal numbers in the FROM and TO fields, you must enter X in this field. When you specify alphabetic values in the FROM and TO fields, you must enter A in this field. The default is a decimal. This property is not case-sensitive.	Optional	String
validationType	Specifies the type of validation. The values for validationType can be SG (Storage Group) or * (Entire SCDS). If the query parameter <i>validate</i> is "Y" or the <i>activate</i> is "Y", the validationType is required. Otherwise, it can be optional.	Optional	String

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

The user’s z/OS user ID must have UPDATE access to the following resource profiles:

- <SAF-prefix>.ZOSMF.STORAGE.SG.VOLUME in the ZMFAPLA class.
- <SAF-prefix>.ZOSMF.STORAGE.SCDS in the ZMFAPLA class to validate or activate the updated SCDS.
- MVS.SETSMS.SMS in the OPERCMDS class to activate the updated SCDS.

The user must have access to z/OS Operator Consoles to activate the updated SCDS.

The user must also have write access for the SCDS.

HTTP status codes

For a successful request, HTTP status code 201 is returned.

For a list of status codes, see [“Error handling” on page 548](#).

Response content

On completion, the z/OS storage management REST interface returns an HTTP response. The same volume serial number can only be added once.

If the response is successful, an object, which contains new volumes is returned.

```
{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ]
}
```

If the response is not successful, a JSON object is returned with additional information.

```
{
  "rc": 8,
  "reason": 6003,
  "category": 4,
  "message": "Dataset not accessed."
}
```

If validation or activation are invoked in the request but fail, volumes that are successfully added are included in the response. If activation results are not immediately available, the HTTP response code is 200 and *activateResURI* and *activateResURL* are returned. You can invoke the activation result URL to get the activation result after the initial response is returned. For more information, see [“Get an SCDS activation result”](#) on page 638.

For the meanings of the category, return code, and reason code fields, see [“Error reporting categories”](#) on page 639.

Example HTTP interaction

1. The example in Figure 271 on page 632 shows the request body to define a volume list to a storage group where *validate* is N and *activate* is N.

```
POST /zosmf/storage/rest/<version>/storagegroups/<stg-name>/volumes?validate=N&activate=N
{
  SCDS: XX.XX.SCDs,
  range:[{
    prefix: "vol",
    from: "1",
    to: "2"
  }],
  validateType: "SG"
}
```

Figure 271. Sample request to define a volume list to a storage group without invoking activation or validation

The following is the response body for the request when adding volumes is successful.

```
{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ]
}
```

Figure 272. Sample response body when adding volumes is successful

The following shows the response body for the request when adding volumes fails.

```
{
  "rc": 8,
  "reason": 6003,
  "category": 4,
  "message": "Dataset not accessed."
}
```

Figure 273. Sample response body when adding volumes fails

2. The example in Figure 274 on page 633 shows the request body to define a volume list to a storage group where *validate* is Y and *activate* is N.

```

POST /zosmf/storage/rest/<version>/storagegroups/<stg-name>/volumes?validate=Y&activate=N
{
  SCDS: XX.XX.SCDs,
  range:[{
    prefix: "vol",
    from: "1",
    to: "2"
  }],
  validateType: "SG"
}

```

Figure 274. Sample request to define a volume list to a storage group and validate the SCDS

The following is the response body for the request when both adding volumes and validation are successful.

```

{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ]
}

```

Figure 275. Sample response body when adding volumes is successful and validation is successful

The following is the response body for the request when the volumes are added successfully, but validating the SCDS fails. The volumes field is returned only when an error occurs during validation or activation, displaying that the list is still added to the SCDS successfully.

```

{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ],
  "rc": 8,
  "reason": 6080,
  "category": 4,
  "message": "Data set is in use."
}

```

Figure 276. Sample response body when adding volumes is successful but validation fails

3. The example in [Figure 277](#) on [page 633](#) shows the request body to define a volume list to a storage group where *validate* is Y and *activate* is Y.

```

POST /zosmf/storage/rest/<version>/storagegroups/<stg-name>/volumes?validate=Y&activate=Y
{
  SCDS: XX.XX.SCDs,
  range:[{
    prefix: "vol",
    from: "1",
    to: "2"
  }],
  validateType: "SG"
}

```

Figure 277. Sample request to define a volume list to a storage group

The following is the response body for the request when the HTTP response code is 200 and *activateRes* is returned, the SCDS is activated successfully.

```
{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ],
  "activateRes": "IGD008I NEW CONFIGURATION ACTIVATED FROM SCDS
                  ZOSMF.DGQ.SCDs BY CONS TEST\r\n IGD403I CURRENT SYSTEM CONFIGURATION\r\n ACTIVATION LEVEL: 120\r\n ACTIVATION
TIME(UTC):      04:15:55.96\r\n ACTIVATION DATE: 2022/07/04\r\n ACTIVATION SOURCE: SETSMS"
}
```

Figure 278. Sample response body when adding volumes and activation is successful

The following is the response body for the request when the HTTP response code is 200 and *activateResURI* and *activateResURL* are returned, the activation result is not available yet. You can invoke the activation result URL to get the activation result later.

The content should be replaced as the following:

```
{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ],
  "activateResURI": "/zosmf/storage/rest/v1/cds/ZOSMF.DGQ.SCDs?console-name=test&Ckey-number=C1405322",
  "activateResURL": "https://pkstp01.pok.stglabs.ibm.com:33028/zosmf/storage/rest/v1/cds/ZOSMF.DGQ.SCDs?console-
name=test&Ckey-number=C1405322"
}
```

Figure 279. Sample response body when the activation result is not available

The following is the response body when adding the volumes is successful, but activation step fails. The volumes field is returned only when an error occurs during validation, displaying that the list is still added to the SCDS successfully.

```
{
  "volumes": [
    {
      "volumeSerial": "vol1"
    },
    {
      "volumeSerial": "vol2"
    }
  ],
  "rc": 1,
  "reason": 9999,
  "category": 5,
  "message": "IGD414I SMS DETECTED count DUPLICATE VOLUME(S) WITHIN THE SMS CONFIGURATION. FIRST DUPLICATE VOLUME IS VOL1C
ATTEMPTING TO REPAIR ACDS"
}
```

Figure 280. Sample response body when validation or activation fails

Validate an SCDS

Use this operation to validate an SMS source control data set.

HTTP method and URI path

```
PUT /zosmf/storage/rest/<version>/cds/<scds-name>
```


Query parameters

Table 360. Query parameters for retrieving a list of volumes

Query Parameter	Description	Required or Optional	Type
scds-name	Specifies the name of the source control data set. The SCDS name can be up to 42 characters, including delimiters (periods). This property is not case-sensitive.	Required	String

Description

This operation validates a specified SMS source control data set.

On successful completion, the request returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success.

Request content

The request content is expected to contain a JSON object. See [Table 361 on page 635](#).

Table 361. Request content for the validate SCDS request

Field name	Description	Required or Optional	Type
validationType	Specifies the type of validation used, either <i>SG</i> for Storage Group, <i>MC</i> for Management Class, <i>SC</i> for Storage Class, <i>DC</i> for Data Class or <i>*</i> for the entire SCDS.	Required	String

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

The user’s z/OS user ID must have UPDATE access to the following resource profiles to validate the updated SCDS:

- `<SAF-prefix>.ZOSMF.STORAGE.SCDS` in the ZMFAPLA class.

The user must also have write access of SCDS.

HTTP status codes

For a successful request, HTTP status code 204 is returned.

For a list of status codes, see [“Error handling” on page 548](#).

Response content

On completion, the z/OS storage management REST interface returns an HTTP response.

If the response is successful, no response is returned.

If the response is not successful, a JSON object will be returned with additional information.

```
{
  "rc": 8,
  "reason": 6003,
  "category": 4,
```

```
}    "message": "Dataset not accessed."
}
```

For the meanings of the category, return code, and reason code fields, see [“Error reporting categories”](#) on page 639.

Activate an SCDS

Use this operation to activate an SMS source control data set.

HTTP method and URI path

```
PUT /zosmf/storage/rest/<version>/cds/<scds-name>
```

Query parameters

Table 362. Query parameters to activate an SCDS request			
Field name	Description	Required or Optional	Type
scds-name	Specifies the name of the source control data set. The SCDS name can be up to 42 characters, including delimiters (periods). This property is not case-sensitive.	Required	String

Description

This operation activates a specified SMS source control data set.

On successful completion, the request returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success.

Request content

None.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

The user’s z/OS user ID must have UPDATE access to the following resource profiles:

- <SAF-prefix>.ZOSMF.STORAGE.SCDs in the ZMFAPLA class to activate the updated SCDS.
- MVS.SETSMS.SMS in the OPERCMDS class to activate the updated SCDS.

The user must have access to z/OS Operator Consoles to activate the updated SCDS.

The user must also have write access to the SCDS.

HTTP status codes

For a successful request, HTTP status code 200 is returned.

For a list of status codes, see [“Error handling”](#) on page 548.

Response content

On completion, the z/OS storage management REST interface returns an HTTP response.

If the response is successful and the response `activateRes` is returned, the response indicates that the activation is successful. If the response code is greater than or equal to 400, the activation failed and the response body will contain the failed reasons. If the response code is 200 and the response body contains the `activateResURI` and `activateResURL`, it indicates that the activation result is inconclusive. You will need to use the GET activation result method to retrieve the result. For more information, see [“Get an SCDS activation result” on page 638](#).

For the meanings of the category, return code, and reason code fields, see [Table 369 on page 657](#).

Example HTTP responses

1. The example in [Figure 281 on page 637](#) shows if the status code is 200 and `activateRes` is returned, the SCDS is activated successfully.

```
{
  "activateRes": "IGD008I NEW CONFIGURATION ACTIVATED FROM SCDS
                  ZOSMF.DGQ.SCDS BY CONS TEST\x\n IGD403I CURRENT SYSTEM CONFIGURATION\x
                  ACTIVATION LEVEL: 120\x ACTIVATION TIME(UTC): 04:15:55.96\x ACTIVATION
                  DATE: 2022/07/04\x ACTIVATION SOURCE: SETSMS"
}
```

Figure 281. Sample success response body

2. The example in [Figure 282 on page 637](#) shows if the HTTP response code is 200 and `activateResURI` and `activateResURL` are returned, it means the activation result is not available yet. You can invoke the activation result URL to get the activation result later. For more information, see [“Get an SCDS activation result” on page 638](#).

```
{
  "activateResURI": "/zosmf/storage/rest/v1/cds/ZOSMF.DGQ.SCDS?console-name="test"&Ckey-number="C1405322",
  "activateResURL": "https://pkstp01.pok.stglabs.ibm.com:33028/zosmf/storage/rest/v1/cds/ZOSMF.DGQ.SCDS?console-name="test"&Ckey-number="C1405322"
}
```

Figure 282. Sample failed response body

3. The example in [Figure 283 on page 637](#) shows that otherwise, an error report will be returned as below.

```
{
  "rc": 1,
  "reason": 9999,
  "category": 5,
  "message": "IGD414I SMS DETECTED count DUPLICATE VOLUME(S) WITHIN THE SMS CONFIGURATION. FIRST DUPLICATE VOLUME IS VOL1C
              ATTEMPTING TO REPAIR ACDS"
}
```

Figure 283. Sample inconclusive response body

Get an SCDS activation result

This operation retrieves the result from an SCDS activation request. The response content contains an HTTP response with a message detailing the result.

HTTP method and URI path

```
GET /zosmf/storage/rest/<version>/cds/<scds-name>
```

Query parameters

Table 363. Query parameters to activate an SCDS request			
Field name	Description	Required or Optional	Type
scds-name	Specifies the name of the source control data set. The SCDS name can be up to 42 characters, including delimiters (periods). This property is not case-sensitive.	Required	String
console-name	The name of the console used to activate SCDS. This value does not need to be specified by the user. It is returned by the previous Activate an SCDS API request. The value must meet the following criteria: <ul style="list-style-type: none">Contains 2 - 8 alphanumeric characters.Begins with an alphabetic or national character. National characters are \$, #, and @.Cannot be defcn, which is reserved.	Required	String
Ckey-number	Specifies the command response key from the activate SCDS request returned by the previous Activate an SCDS API. The user does not need to specify this value. The value must meet the following criteria: <ul style="list-style-type: none">Begins with c C.Contains of 7 numeric digits of 0-9.	Required	String

Description

On completion, the request returns the result of the specified activation request. The response body indicates if the activation was successful, failed or could not be found.

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 3.

The user must have access to z/OS Operator Consoles to get the activation result.

HTTP status codes

For a successful request, HTTP status code 200 is returned.

For a list of status codes, see [“Error handling”](#) on page 548.

Response content

On completion, the z/OS Storage Management REST interface returns an HTTP response.

If the response code is 200 and the response `activateRes` is not empty, the response indicates that the SCDS has been activated successfully. If the response code is greater than or equal to 400, the activation failed and the response body will contain a description of the reasons that it failed.

If the response code is 200 and the response `activateRes` is empty, either the activation result has been retrieved before or the Ckey-number does not exist.

Example HTTP responses

1. The example in [Figure 284 on page 639](#) shows when the HTTP response code is 200 and `activateRes` is returned, meaning the SCDS is successfully activated.

```
{
  "activateRes": "IGD008I NEW CONFIGURATION ACTIVATED FROM SCDS
ZOSMF.DGQ.SCDS BY CONS TEST\r\n IGD403I CURRENT SYSTEM CONFIGURATION\r
ACTIVATION LEVEL: 120\r ACTIVATION TIME(UTC): 04:15:55.96\r ACTIVATION
DATE: 2022/07/04\r ACTIVATION SOURCE: SETSMS"
}
```

Figure 284. Sample success response body

2. The example in [Figure 285 on page 639](#) shows a failed result where HTTP response code is 400, and a report is returned.

```
{
  "rc": 1,
  "reason": 9999,
  "category": 5,
  "message": "IGD414I SMS DETECTED count DUPLICATE VOLUME(S) WITHIN THE SMS CONFIGURATION. FIRST DUPLICATE VOLUME IS VOL1C
ATTEMPTING TO REPAIR ACDS"
}
```

Figure 285. Sample response body

Error reporting categories

This section describes the error categories and associated error codes that can be returned in the JSON error report document for storage management API services requests.

Categories

Table 364 on page 639 shows the error categories that are defined for errors that are returned in storage management API services operations.

Table 364. Error categories for storage management API services operations			
Category	Ordinal Value	Description	Where the error details are described
Service error	1	Errors that are produced or detected in the service layer.	“Category 1 — Service error” on page 640
SSI extended status	2	Errors that are produced or detected by the extended status function call of the subsystem interface (SSI Function Code 55).	Table 366 on page 643

Table 364. Error categories for storage management API services operations (continued)

Category	Ordinal Value	Description	Where the error details are described
Unexpected	3	Unexpected errors detected.	Table 367 on page 644
SCDS modification and validation error	4	Errors that are produced or detected in SCDS modification and validation.	Table 368 on page 645
SCDS activation error	5	Errors that are produced or detected in SCDS activation.	Table 369 on page 657
Security check error	6	Errors that are produced or detected in security check.	Table 370 on page 657

Category 1 – Service error

“Category 1 – Service error” on page 640 shows the possible conditions for this error category.

Table 365. Category 1 errors for storage management API services operations

rc	reason	message	Description
8	1	Value of "%s" query parameter is not valid.	<p>%s can be a filter, stg-name, limit, or offset.</p> <p>Filter is a volume serial number filter to filter volumes.</p> <p>The volume serial number (volume-ser) can be:</p> <ul style="list-style-type: none"> • A fully specified volume serial number, one to six characters • A partially specified volume serial number that uses an asterisk (*) as a placeholder for characters. • A single asterisk (*) for all mounted volumes. <p>stg-name is storage group name, case-insensitive. This value is 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.</p> <p>The limit is the maximum number of volumes that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>The request retrieves volumes starting from the offset value. The valid range of offset values is 0-2147483647. The offset cannot be larger than or equal to the number of volumes.</p>
8	2	Value of "%s" query parameter is not valid.	<p>%s is The volume serial number (volume-ser) for a Get Volume Definition request.</p> <p>The volume serial number is a fully specified volume serial number, one to six characters long.</p>

Table 365. Category 1 errors for storage management API services operations (continued)

rc	reason	message	Description
8	3	Value of "%s" query parameter is not valid.	<p>%s can be a filter, type, volume-ser, limit, or offset for a Get Storage Group List request.</p> <p><i>storage group name</i> filter is 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character, or a partially specified storage group name by using an asterisk (*)</p> <p><i>storage group type</i> can be VIO, Pool, Dummy, Copy Target, Object, Object Backup, or Tape.</p> <p>The volume serial number (volume-ser) is a fully specified volume serial number, one to six characters long.</p> <p>The limit is the maximum number of storage groups that can be retrieved. The valid range of limit values is 0 - 2147483647.</p> <p>The request retrieves storage groups starting from the offset value. The valid range of offset values is 0-2147483647. The offset cannot be larger than or equal to the number of volumes.</p>
8	4	Value of "%s" query parameter is not valid.	<p>%s can be <i>stg-name</i> for a GET Storage Group Definition request.</p> <p><i>stg-name</i> is 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character.</p>
8	5	No match for method GET and pathInfo='pathInfo'.	URL for the GET request is not correct.
8	6	The requested method %s is not supported.	%s is not the GET method.
8	7	The value of offset "%d" is equal to or greater than the "%s" count "%t".	<ul style="list-style-type: none"> • %d is the offset value. • %s can be storage groups or volumes. • %t is the actual storage groups or volumes count. <p>The offset cannot be larger than or equal to the number of storage groups or volumes.</p>
8	8	The "%s" cannot be used with the filter parameter in the same request.	%s is the offset or limit. The offset or limit cannot be used with the filter parameter in the same request.

Table 365. Category 1 errors for storage management API services operations (continued)

rc	reason	message	Description
8	9	Value of "%s" query parameter is not valid.	<p>%s can be <i>filter</i> for a GET Data Class list request. <i>filter</i> is 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character. The asterisk (*) and percent (%) wildcards are allowed.</p> <ul style="list-style-type: none"> • Asterisk matches any number of characters. • Percent matches one character. • Asterisk and percent wildcards can be used together. • You can use more than one asterisk and more than one percent wildcard.
8	10	Value of "%s" query parameter is not valid.	<p>%s can be <i>filter</i> for a GET Storage Class list request. <i>filter</i> is 1 - 8 alphanumeric characters, starting with an alphabetic or special (\$, @ or #) character. The asterisk (*) and percent (%) wildcards are allowed.</p> <ul style="list-style-type: none"> • Asterisk matches any number of characters. • Percent matches one character. • Asterisk and percent wildcards can be used together. • You can use more than one asterisk and more than one percent wildcard.
8	11	The request body is empty.	No request body is provided in REST APIs.
8	12	The request body is not a valid JSON body.	The requested body is not a valid JSON body.
8	13	The parameter %s is missing or the value is invalid.	You must provide the valid query or body parameters according to the REST API description.
8	14	The volume serial number must be less than or equal to 6 characters.	The volume serial number can only contain up to 6 characters.
8	15	The FROM and TO fields in the specified range must both have a value and have a valid range.	If you define either the FROM or TO field for your range, you must also define the other. These values must also have a valid range.
8	16	PREFIX or SUFFIX must have a value when FROM or TO is specified.	If you define the FROM or TO fields for your range, you must also define either the PREFIX or the SUFFIX field.
8	17	The FROM and TO must have a value when SUFFIX is specified.	If you define the SUFFIX field for your range, you must also specify values for the FROM and TO fields. The FROM and TO values must also have the same length.

Table 365. Category 1 errors for storage management API services operations (continued)

rc	reason	message	Description
8	18	Either SUFFIX or PREFIX must have a value when FROM and TO are defined.	If you define the FROM or TO fields for your range, you must also define either the PREFIX or the SUFFIX field.
8	19	If the specified value of TYPE is X, FROM and TO must be defined as hex numbers.	If you define the TYPE value as X, the FROM or TO values must be hex numbers.
8	20	Invalid %s in range.	Valid TYPE values can be A for alphabetic, X for hex numbers, or empty for decimals.
8	21	If the specified value of TYPE is A, the length of FROM or TO should be only one character.	If you define the TYPE value as A, the FROM and TO values should only have a one character length. This character should be alphabetic.
8	22	If the value of TYPE is not specified, the FROM and TO values must be a decimal.	If you do not define a value for TYPE, the default values of FROM and TO must be in decimal format.
8	23	Storage group name %s is invalid.	Provide a valid storage group name that is in the SCDS. The value must meet the following criteria: <ul style="list-style-type: none"> • Contains 1 - 8 alphanumeric characters • Begins with an alphabetic or national character. National characters are \$, #, and @.
8	24	No range property is specified.	Range property is required, it must be specified in body.
8	25	No volume serial can be filtered by the current range.	According to the current range, no volume can be generated to add to the storage group.
8	26	No match for method PUT and pathInfo=%s.	No match was found for method PUT and pathInfo=%s.
8	27	No match for method POST and pathInfo=%s.	No match was found for method POST and pathInfo=%s.
8	28	Volume Information should only be specified to POOL, DUMMY, and COPY POOL BACKUP Storage Groups. The current Storage Group Type is %s.	%s is the current storage group Type. Only storage groups with POOL, DUMMY, or COPY POOL BACKUP type can be added volumes.

Category 2 – SSI extended status

“Category 2 – SSI extended status” on page 643 shows the possible conditions for this error category.

Table 366. Category 2 errors for storage management API services operations

rc	reason	message
4	0004	An invalid SSSA1D field was specified.
4	0008	An invalid SSSAVER field was specified.

Table 366. Category 2 errors for storage management API services operations (continued)

rc	reason	message
4	0012	An unknown SMS service was requested.
4	0016	SMS does not support the specified SSI function code.
4	0020	The caller is not authorized to perform the requested SMS function.
4	0300	An SMS abend occurred.
4	0301	The data set is not eligible for allocation on SMS-managed volumes.
4	0302	The storage class is incompatible with the volume.
4	0303	A data set that is eligible for SMS management was made ineligible.
4	0306	An error occurred during SMS processing; the error was logged.
4	0307	An error occurred in an ACS installation exit.
4	0308	The caller is not authorized to the specified SMS class.
4	0309	An SMS data set is not allowed within the scope of a JOBCAT/ STEPCAT.
4	0310	Management class nor storage class cannot be specified for a data set that is not eligible to be managed by SMS.
4	5000	An invalid value for SSSA1TYP was specified.
4	5001	The caller passed insufficient storage.
4	5002	The construct was not found.
4	5003	No volumes are associated with the storage group.
4	5005	Data could not be returned to the area that was provided by the caller.
4	5007	The volume definition in the active configuration is associated with a nonexistent storage group on a SSSA1TYP=8 request.
4	5008	The count value (SSSA1CNT) must be greater than 0.
4	5009	Unable to acquire sufficient storage for the caller.

Category 3 – Unexpected error

Table 367 on page 644 shows the possible conditions for this error category.

Table 367. Category 3 errors for storage management API services operations

rc	reason	message	Description
16	1	Server error occurred.	For details about the exception, check the z/OSMF logs.

Category 4 – SCDS modification and validation error

Table 368 on page 645 shows the possible conditions for this error category.

Table 368. Category 4 errors for SCDS modification and validation error		
rc	reason	message
1	6001	Function code out of range.
1	6002	Invalid action token.
1	6003	Data set not accessed.
1	6004	Data set not formatted.
1	6005	Data set accessed in read mode.
1	6006	Invalid type specified by the caller.
1	6007	Error detected by error recovery.
1	6008	COMMDS must be accessed in update mode.
1	6009	CDS in memory is bad (for example, due to an abend); reaccess or refresh is required.
1	6010	CDS cannot be named "ACTIVE".
1	6030	Data set not found.
1	6031	Caller has insufficient authority.
1	6032	CI size is invalid (not 4K).
1	6033	Access token not initialized.
1	6034	Unable to perform access.
1	6035	Access mode is read; data set is empty.
1	6036	Invalid data set type.
1	6037	A required resource for accessing the data set is unavailable.
1	6038	A list ID field is invalid.
1	6039	DIV was unable to process the VSAM SHAREOPTIONS.
1	6040	Data set previously updated.
1	6041	Unable to perform save.
1	6042	Data set size has changed (COMMDS).
1	6043	Reserve required and not held.
1	6044	Same active ACDS specified for message IGD069
1	6050	Data set is invalid. Run VALIDATE.
1	6051	Unable to perform unaccess.
1	6052	CDS is in 8-name mode and the SMSplex is in 32 name mode; conversion was not requested.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6053	The CDS is in 32-name mode and the SMSplex is in 8-name mode.
1	6054	Save failed; another system converted the configuration or COMMDS to 32-name mode.
1	6055	No systems defined to match the system name.
1	6056	CDS formatted at higher level, operation failed. Edit to CDS that is formatted at higher release need to be unaccessed. Retry the operation on system at the formatted or higher level.
1	6058	Device services failed to return UCB for DD.
1	6059	Data set size changed Indicates high access activity.
1	6060	Unable to perform reserve.
1	6061	Unable to perform release.
1	6062	SVC99 error; error detected by SMS.
1	6063	SVC99 error; dynamic allocation or unallocation.
1	6064	Unable to find UCB to perform reserve or release.
1	6065	SVC99 parameter list is invalid.
1	6066	Environmental error prevented dynamic allocation or the operation in progress.
1	6067	DIV failure; undeterminable failure.
1	6068	DIV failure; system service failed (called by DIV). Problem could be caused by insufficient space to extend the data set. See the system console for IEC messaged that might be related to the error.
1	6069	DIV failure; I/O error. See the system console for IEC messages that might be related to the error.
1	6070	No matching elements found.
1	6071	Element not found.
1	6072	Element was replaced.
1	6073	Invalid item count; must be 1.
1	6074	Invalid item length.
1	6075	Insufficient storage.
1	6080	Data set is in use.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6081	ENQ failed for data set.
1	6082	DEQ failed for data set.
1	6090	ACS routine not found.
1	6091	Data set is invalid.
1	6092	BCDEFLEN is zero.
1	6093	BCDEFLEN is bad.
1	6094	Extension length is zero.
1	6095	Extension length is bad.
1	6096	Extension offset is zero.
1	6097	Extension offset is bad.
1	6098	Extension offset is out of ascending order.
1	6099	Extension lengths plus definition length do not add up to total length specified in header.
1	6101	Function not allowed for the active configuration.
1	6102	Subsystem not active; no active configuration available.
1	6103	Error detected by SSI.
1	6104	Access to active configuration not allowed in update mode.
1	6105	Conditional GETMAIN failed.
1	6110	System or system group name not found.
1	6111	System name or group name not found.
1	6112	System/system group name list is full.
1	6113	No base configuration information found.
1	6114	System name type specified is invalid.
1	6130	Data set is not a linear data set.
1	6131	Access failed; the data set type is incorrect. The referenced data set was found to be a different type (ACDS, SCDS, COMMDS) than the type requested.
1	6132	COPYSCDS command input SCDS dataset is an ACDS dataset.
1	6133	COPYSCDS command input SCDS dataset is not formatted.
1	6134	COPYSCDS command input SCDS dataset is invalid.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6135	COPYSCDS command target dataset is ACTIVE ACDS or is same as the source dataset.
1	6136	Your configuration data set cannot be an extended format linear data set.
1	6700	Object or object backup storage group connected to system group.
1	6701	Object or object backup storage group connected to more than one system.
1	6702	Optical drive connected to system group.
1	6703	Optical drive connected to more than one system.
1	6704	SG: Breakpoint value invalid.
1	6705	SG: Track Alloc Threshold value invalid.
1	6706	SC: OAM Enhancement error.
1	6707	Duplicate volume found in target CDS.
1	6708	Invalid alert threshold value specified.
1	6709	Serialization error - invalid action.
1	6710	Processing priority invalid.
1	6711	Invalid RMODE31 value specified.
1	6712	MC: Invalid Volume Set Management Level Value.
1	6713	MC: Invalid Retention Method Value.
1	6714	MC: If Retention Method is set to VRSEL, the fields "Volume Set Management Level" and "Retain While Cataloged" should not be set.
1	6715	MC: If Retention Method is set to EXPDT, the field "Exclude from VRSEL" should not be set.
1	6732	Invalid DS recovery value.
1	6733	Invalid PPRC recovery value.
1	6734	Invalid PPRC back value.
1	6735	Invalid HFRR value.
1	6736	SGCL: No SG in collection.
1	6737	SGCL: Catalog list maximum exceeded.
1	6738	SGCL: Catalog name syntax incorrect.
1	6739	SGCL: SG array not sorted.
1	6740	Unknown storage group collection type SGCTYPE.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6741	COPYPOOL name contains an incorrect length.
1	6742	COPYPOOL name contains an incorrect syntax.
1	6743	Maximum number of copy pool storage groups exceeded.
1	6744	A storage group name contains an incorrect length.
1	6745	A storage group name contains an incorrect syntax.
1	6746	Incorrect number of versions specified.
1	6747	No SGs were specified for this storage group collection.
1	6748	COPYPOOL BACKUP storage group name contains incorrect syntax or length.
1	6749	Only POOL storage groups can specify a COPYPOOL BACKUP storage group.
1	6750	Invalid DS separation profile name.
1	6751	Invalid OVERFLOW value.
1	6752	Invalid extend SG name.
1	6753	Internal SMS parm list IGDIOPL contains incorrect information.
1	6754	DS separation profile record format is not fixed.
1	6755	Open of DS separation profile failed.
1	6756	Space constraint relief must also be specified when maximum volume count is specified.
1	6757	Invalid maximum volume count value.
1	6758	SGCL: Missing dump class when auto dump is ON.
1	6759	SGCL: Dump sys or sys group name syntax is incorrect.
1	6760	SGCL: Dump class name syntax is incorrect.
1	6761	Base: Lock set length is invalid.
1	6762	Base: Lock set name is invalid.
1	6763	Base: Lock structure length invalid.
1	6764	Base: Lock structure name invalid.
1	6765	Base: Lock set count invalid.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6766	Base: Lock set point is 0.
1	6767	Base: Lock structure list is out of order.
1	6768	Base: Lock structure index is invalid.
1	6769	Base: Lock structure pointer is 0.
1	6770	SC: Lock set length is invalid.
1	6771	SC: Lock set name is invalid.
1	6772	DC: Invalid media type or recording technique for key.
1	6773	DC: Key label and key code set are mismatched. Keylabel1 is specified without specifying keycode1.
1	6774	DC: Invalid key label length. Maximum length is 64 characters.
1	6776	DC: Invalid key code value. Must be 'L' or 'H'.
1	6777	DC: SMBVSP value invalid.
1	6778	DC: Override without SPACE.
1	6779	DC: EATTR value invalid.
1	6780	Separation profile unexpected end of file. Profile was modified during SMS configuration activation.
1	6781	Separation profile number of sep groups mismatch. Profile was modified during SMS configuration activation.
1	6782	Separation profile number of sep group DSNs mismatch. Profile was modified during SMS configuration activation.
1	6783	Error accessing data set separation profile. See message IGD06031I.
1	6800	DC: NAME invalid.
1	6801	RECORD is invalid.
1	6802	RECFM is invalid.
1	6803	Carriage control is invalid.
1	6804	LRECL is invalid.
1	6805	Key length is invalid.
1	6806	Key off is invalid.
1	6807	Retention period is invalid.
1	6808	Expiration day/date is invalid.
1	6809	Primary space is invalid.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6810	Secondary space is invalid.
1	6811	Directory blocks space is invalid.
1	6812	Volume count is invalid.
1	6813	CI size is invalid.
1	6814	CI free percent is invalid.
1	6815	CA free percent is invalid.
1	6816	VSAM share options is invalid.
1	6817	Name is invalid for management class.
1	6818	Minimum days on primary is invalid.
1	6819	Retain days archival is invalid.
1	6820	Expiration format is invalid.
1	6822	Backup frequency is invalid.
1	6823	Backup versions is invalid.
1	6824	Backup retention is invalid.
1	6825	Backup retention/no primary is invalid.
1	6826	Maximum retention is invalid.
1	6827	Name is invalid for storage class.
1	6828	Millisecond response is invalid.
1	6829	Sequential kilobyte response is invalid.
1	6830	Name is invalid for storage group.
1	6831	Storage group type is invalid.
1	6832	VIOMAX is invalid.
1	6833	VIUNIT is invalid.
1	6834	System names are invalid.
1	6835	System statuses are invalid.
1	6836	Thresholds are invalid.
1	6837	Default management class is invalid.
1	6838	Volser is invalid.
1	6839	SMS status is invalid.
1	6840	MVS status is invalid.
1	6841	Name length is invalid.
1	6842	Unknown storage group for volume.
1	6843	Storage group is type VIO, OBJECT, or OBJECT BACKUP or TAPE.
1	6844	Volser length is not equal to 6.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6845	Duplicate system names found.
1	6846	Allocation unit less is than 1.
1	6847	Scale factor (AVGREC) invalid.
1	6848	Availability is invalid.
1	6849	Direct response is invalid.
1	6850	Direct bias is invalid.
1	6851	Sequential response is invalid.
1	6852	Sequential bias is invalid.
1	6853	Level 1 days is invalid.
1	6854	Command/automatic migrate is invalid.
1	6855	Number of GDG elements is invalid.
1	6856	Data set deleted; backup version is invalid.
1	6857	Administrator/end user backup is invalid.
1	6858	RECORD and RECFM are mutually exclusive.
1	6859	Autobackup system is invalid.
1	6860	Autodump system is invalid.
1	6861	Automigrate system is invalid.
1	6862	Reason code is no longer used.
1	6863	Reason code is no longer used.
1	6864	Reason code is no longer used.
1	6865	Minimum days on primary is required.
1	6866	Level 1 days required.
1	6867	Backup versions are required.
1	6868	Backup versions (data set delete) are required.
1	6869	Backup frequency is required.
1	6870	Backup retention is required.
1	6871	Retain days only backup is required.
1	6872	MCPREL and MCPRECN are mutually exclusive.
1	6873	Incorrect media type specified with performance scaling.
1	6874	Scaling and segmentation are mutually exclusive.
1	6875	Incorrect media type specified with performance segmentation.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6880	DSNTYPE is invalid.
1	6881	RECORG and DSNTYPE are mutually exclusive.
1	6882	Backup destination class is invalid.
1	6883	Expiration action is invalid.
1	6884	Retain days archive is invalid.
1	6885	Name is invalid.
1	6886	Expiration date is invalid.
1	6887	Destination is invalid.
1	6888	Data set prefix is invalid.
1	6889	Instruction DSN is invalid.
1	6890	Instruction member is invalid.
1	6891	Data set name is invalid.
1	6892	Data set member invalid.
1	6893	Destination is required.
1	6894	Output prefix is required.
1	6895	Instruction DSN is required.
1	6896	One DSN is required.
1	6897	Guaranteed backup frequency is invalid.
1	6898	Guaranteed backup frequency is required with AUTOBACKUP=Y.
1	6899	Invalid block size limit value.
1	6900	Time Since Creation value is invalid.
1	6901	Time Since Last Use value is invalid.
1	6902	Periodic value is invalid.
1	6903	Group ID is not unique.
1	6904	Object Cycle Start is invalid.
1	6905	Object Cycle End is invalid.
1	6906	Initial Access is invalid.
1	6907	Library name is invalid.
1	6908	Library type is invalid.
1	6909	Drive name is invalid.
1	6910	Drive type is invalid.
1	6911	Time Since Creation and Time Since Last Use fields are mutually exclusive.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6912	Time Since Creation and Periodic fields are mutually exclusive.
1	6913	Time Since Last Use and Periodic fields are mutually exclusive.
1	6914	Periodic Monthly and Periodic Quarterly fields are mutually exclusive.
1	6915	Periodic Monthly and Periodic Yearly fields are mutually exclusive.
1	6916	Periodic Quarterly and Periodic Yearly fields are mutually exclusive.
1	6917	Library status is invalid.
1	6918	Drive status is invalid.
1	6919	Group qualifier is invalid.
1	6920	Duplicate library names in one storage group.
1	6921	Storage Group INTERVAL MIGRATION cannot be selected unless AUTO MIGRATION has also been selected.
1	6922	Copy technique is invalid in management class.
1	6923	Accessibility is invalid in storage class.
1	6924	Library name length is invalid in storage group.
1	6925	Library name is invalid in storage group.
1	6926	Primary space management requested without auto migration in storage group.
1	6927	The library name length is nonzero or the library name is found in a storage that is not type OBJECT, OBJECT BACKUP, or TAPE.
1	6941	The entry default volume use attribute is invalid in library.
1	6942	Eject default is invalid in library.
1	6943	Library ID in library configuration database is invalid in library.
1	6944	Library device type is invalid in library.
1	6945	Entry default unit name is invalid in library.
1	6946	Media type is invalid in data class.
1	6947	Recording technique is invalid in data class.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6948	Compaction type is invalid in data class.
1	6949	Default entry data class is invalid in library.
1	6950	Management class name length is invalid in aggregate group.
1	6951	Management class name is invalid in aggregate group.
1	6952	SES cache length is invalid in the base configuration.
1	6953	SES cache name is invalid in the base configuration.
1	6954	Cache length is invalid in the base configuration.
1	6955	Cache name is invalid in the base configuration.
1	6956	SES cache count is invalid in the base configuration.
1	6957	SES cache pointer is zero in the base configuration.
1	6958	Pointer to aggregate backup parameters is 0 in the management class.
1	6959	Aggregate backup name is invalid in the management class.
1	6960	Aggregate backup name length is invalid in management class.
1	6961	Destination name is invalid in management class.
1	6962	Destination name length is invalid in management class.
1	6965	Indication of number of aggregate backup versions is invalid in management class.
1	6966	Indication of retention period for only version is invalid in management class.
1	6967	Indication of retention period for extra versions is invalid in management class.
1	6968	Copy serialization option is invalid in management class.
1	6969	Retain only version is invalid in management class.
1	6970	Retain extra versions is invalid in management class.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	6971	Number of versions to be backed up is invalid in management class.
1	6972	Destination pointer is zero.
1	6973	Incorrect number of destinations.
1	6974	Destination length is incorrect.
1	6975	Destination name is incorrect.
1	6976	Destination name length is incorrect.
1	6977	Destination name is incorrect.
1	6978	Time is incorrect.
1	6979	NetView® class is incorrect.
1	6980	File encryption is incorrect.
1	6981	Transmit accompany files incorrect.
1	6982	Transmit control information is incorrect.
1	6983	Number of copies is not specified in aggregate group.
1	6984	Number of copies is invalid in aggregate group.
1	6985	Aggregate copy technique is invalid in management class.
1	6990	SES cache list is not in ascending order in base configuration.
1	6991	Cache set name is invalid in storage class.
1	6992	Cache set name length is invalid in storage class.
1	6993	Direct CF weight is invalid in storage class.
1	6994	Sequential CF weight is invalid in storage class.
1	6995	SES cache index is invalid in base configuration.
1	6996	Cache set pointer is zero in base configuration.
1	8000	Cloud construct validation error. The construct name length is invalid.
1	8001	Cloud construct validation error. The construct name syntax is invalid.
1	8002	Cloud construct validation error. The identity is invalid.

Table 368. Category 4 errors for SCDS modification and validation error (continued)

rc	reason	message
1	8003	Cloud construct validation error. Th cloud provider is invalid.
1	8004	Cloud construct validation error. The URI endpoint is invalid.
1	8005	Cloud construct validation error. SSL version is invalid.
1	9998	Unknown error occurred when modifying the SCDS.
1	9999	Unknown error occurred when validating the SCDS.
2	1	Error detected: %s

Category 5 – SCDS Activation error

Table 369 on page 657 shows the possible conditions for this error category.

Table 369. Category 5 errors for activating SCDS action

rc	reason	message	Description
1	1	z/OS console services are not enabled.	z/OS console services are not enabled.
1	9999	*	An error occurred when activating the SCDS. The message contains the command response of the SETSMS command.

For SCDS activation errors, the user can reference [Table 413 on page 770](#).

To get an SCDS activation result, the user can reference [Table 417 on page 783](#).

Category 6 – Security check error

Table 370 on page 657 shows the possible conditions for this error category.

Table 370. Category 6 errors for security check error

rc	reason	message
1	1	The current user does not have update access to the profile <SAF-prefix>.ZOSMF.STORAGE.SCDS in the ZMFAPLA class.
1	2	The current user does not have update access to the profile <SAF-prefix>.ZOSMF.STORAGE.SG.VOLUME in the ZMFAPLA class.
1	3	The current user is not authorized to use the z/OS console services API.
2	1	The current user does not have update access to the profile MVS.SETSMS.SMS in the OPERCMDS class.

Sysplex management services

The sysplex management services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services.

Table 371 on page 658 lists the operations that the sysplex management services provide.

Table 371. Operations provided through the sysplex management services.	
Operation	HTTP method and URI path ¹
“Get a list of CFRM policies” on page 659	GET /zosmf/sysplex/rest/<version>/policies/<policy-type>
“Get CFRM policy properties” on page 661	GET /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
“Activate CFRM policy” on page 667	PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/activation/<pol-name>
“Rename existing CFRM policy” on page 668	PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/name/<pol-name>
“Modify CFRM policy properties” on page 670	PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
“Add CFRM policies to administrative policy” on page 676	POST /zosmf/sysplex/rest/<version>/policies/<policy-type>
“Copy existing CFRM policy with a new name” on page 684	POST /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
“Delete multiple CFRM policies from an administrative policy” on page 685	DELETE /zosmf/sysplex/rest/<version>/policies/cfrm?name=<pol-name>&name=<pol-name2>...

Notes:

1. The valid value for <version> is v1.

Using the Swagger interface

You can use the Swagger interface to display information about the sysplex management services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Required authorizations

The client must be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200
OK

HTTP 201

For a successful creating request, 201 Created (No content) will be returned.

HTTP 204

On successful completion, HTTP status code 204 (No content) is returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 403 Forbidden

The submitter of the request did not authenticate to z/OSMF or the server rejected the request.

HTTP 404 Not found

The target of the request was not found.

HTTP 500 Internal server error

The server encountered an error that prevented it from completing the request.

HTTP 503 Service unavailable

The service is currently unavailable to process the request.

Some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report” on page 686](#).

Error logging

Errors from the sysplex management services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Get a list of CFRM policies

This operation retrieves a list of currently defined administrative CFRM policies.

HTTP method and URI path

```
GET /zosmf/sysplex/rest/<version>/policies/<policy-type>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm

Description

On successful completion, the request returns a list of currently defined administrative CFRM policies.

For a description of the response content, see [“Response content” on page 660](#).

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Required authorizations

The user’s z/OS user ID must have READ access to the following resource profiles:

- *<SAF-prefix>*. ZOSMF.SYSPLEX in the ZMFAPLA class.
- MVSADMIN.XCF.CFRM in FACILITY class to view the administrator CFRM policies.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 658.](#)

If the request is successful, the response also includes a JSON object, where:

items

name

Indicates the name of the CFRM policy.

_user

Indicates the user ID.

_defined

Indicates the date and time of modification of the CFRM policy.

activePolicy

name

Indicates the name of the activated CFRM policy.

activatedDate

Indicates the date and time the CFRM policy is activated.

_json_version

4

Example

In the following example, the GET method is used to retrieve a list of the currently defined administrative CFRM policies.

```
{
  "items": [
    {
      "name": "ALTERPOL",
      "user": "IBMUSER",
      "_defined": "02/25/2020T12:04:56.187608Z"
    },
    {
      "name": "CTTEST1",
      "user": "IBMUSER",
      "_defined": "02/25/2020T12:04:56.187608Z"
    },
    {
      "name": "REBLDPCT",
      "user": "IBMUSER",
      "_defined": "02/25/2020T12:04:56.187608Z"
    },
    {
      "name": "UMDUPLX",
      "user": "IBMUSER",
      "_defined": "02/25/2020T12:04:56.187608Z"
    }
  ],
  "activePolicy": {
    "name": "CTTEST1",
    "activatedDate": "02/25/2020T16:09:18.362777Z"
  },
  "_json_version": "4"
}
```

Figure 286. Sample response from a request to retrieve a list of the currently defined administrative CFRM policies.

Get CFRM policy properties

This operation retrieves the properties of a CFRM policy.

HTTP method and URI path

```
GET /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm
- **pol-name** is the URI path variable that identifies the name of a policy.

Description

On successful completion, the request returns a list of the CFRM policy properties.

For a description of the response content, see [“Response content” on page 662](#).

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

The user’s z/OS user ID must have READ access to the following resource profiles:

- <SAF-prefix>. ZOSMF.SYSPLEX in the ZMFAPLA class.
- MVSADMIN.XCF.CFRM in FACILITY class to view the administrator CFRM policies.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 658.

If the request is successful, the response also includes a JSON object, where:

data type

CFRM

_version_supported

Supported version

policy

name

Indicates the name of the CFRM policy.

_defined

Indicates the date and time of modification of the CFRM policy.

_user

Indicates the user ID.

_version

Indicates the data version.

CF

Specifies the definition of a coupling facility within the scope of the named policy. The location of the information that is needed to define the coupling facility in a CFRM policy depends on the processor on which the coupling facility is installed.

The limit for the number of coupling facilities that can be defined in a policy is established when the CFRM couple data set is formatted.

The coupling facility must appear in the preference list of at least one coupling facility structure that is defined within the policy or the administrative policy utility flags the statement as an error.

NAME

Specifies the 1 - 8 character length name of the coupling facility. The valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), national characters (\$,@,#), or an underscore (_). The **cfname** must start with an alphabetic character (A-Z).

Consider defining your coupling facility name to match the name of the LPAR in which the coupling facility is to run.

NAME is a required parameter.

The following set of parameters describe the unique coupling facility that is defined in the policy.

TYPE

Specifies the 6-character machine type. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. TYPE is a required parameter.

MFG

Specifies the 3-character manufacturer identification. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). MFG is a required parameter.

PLANT

Specifies the 2-character plant of manufacture code. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). PLANT is a required parameter.

SEQUENCE

Specifies the 12-character serial number. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. SEQUENCE is a required parameter.

PARTITION

Specifies the 1- or 2-hexadecimal digit qualifier to uniquely identify and associate a coupling facility to a specific PR/SM partition.

- If the coupling facility is defined on a System z9®, z990, and z890 processor, then the PARTITION value must match the partition ID associated on the activation profile for the CF image on the support element or hardware master console.
- If the coupling facility is defined on a pre-z990 or pre-z890 or earlier processor, then the PARTITION value must match the partition number specified by HCD/IOCP in the IOCDs for that partition and can be in the range of 01-0F.
- Otherwise, the PARTITION value can be in the range of 0-F and is the same as the partition number defined in HCD.

Partition is a required parameter.

SIDE

Specifies the 1-numeric character to identify the coupling facility to a specific physical side of a CPC running in Physically Partitioned mode. You should not split or merge physical sides on which a coupling facility resides.

Use caution when splitting or merging physical sides of a processor that contains a coupling facility.

SIDE is an optional parameter. Its omission implies that the coupling facility resides on a single image CPC.

- SIDE(0) — Side 0 of a physically partitioned machine. Do not specify for a Single Image CPC.
- SIDE(1) — Side 1 of a physically partitioned machine. Do not specify for a Single Image CPC.

CPCID

Specifies the 2-hexadecimal digit qualifier (00-FF) to identify a coupling facility to a specific CPC in a CMOS processor. CPCID is an optional parameter.

DUMPSPACE

Specifies the amount of space to be reserved in the coupling facility for dumping structures allocated in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1-10 decimal digits long. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

DUMPSPACE is an optional parameter. If omitted, no storage in the coupling facility is reserved for dumping.

SITE

Specifies the site at which the coupling facility resides. When specified, this information (along with status information from a Recovery Manager (for example, GDPS®)), will be used in break duplexing decisions to keep the structure instance at the recovery site. SITE changes take effect when the policy is activated.

This keyword should be considered when specifying the DUPLEX **dupsite** and **dupmode** parameters as part of the STRUCTURE definition.

STRUCTURE

Specifies the definition of a structure within the scope of the named policy. The limit for the number of structures that can be defined in a policy is established when the CFRM couple data set is formatted.

NAME

Specifies the 1-16 character name of the structure. The valid characters are numeric characters, uppercase alphabetic characters, national characters (\$,@,#), or an underscore (_). The *strname* must start with an alphabetic character (A-Z). IBM names begin with SYS, or the letters A through I.

Structures that are used for XCF signaling must begin with the letters IXC.

NAME is a required parameter.

SIZE

Specifies the maximum amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long.

SIZE is a required parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

INITSIZE

Specifies the initial amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number must be 1 - 10 decimal digits long. The INITSIZE value must be less than or equal to the SIZE value. Otherwise, the system issues error message IXC745I.

INITSIZE is an optional parameter. If not specified, the system uses the SIZE parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

MINSIZE

Specifies the smallest size to which the structure can ever be altered, preventing allocation of an unusable structure. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long. The MINSIZE value must be less than or equal to the INITSIZE value, or less than or equal to the SIZE value if INITSIZE is not specified.

MINSIZE is an optional parameter. The largest size that can be specified is 1T. Sizes larger than 1T will cause message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value. If not specified and ALLOWAUTOALT(NO) is specified or defaulted to, the MINSIZE used will be 0. If not specified and ALLOWAUTOALT(YES) is specified, the system will set MINSIZE to 75% of INITSIZE, or to 75% of SIZE if INITSIZE is not specified.

SCMMAXSIZE

Specifies the maximum amount of storage-class (“flash”) memory that is assignable for use by this structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. Valid values for the number are 1 - 11 decimal digits long and greater than 0, if this keyword is specified.

SCMMAXSIZE is an optional parameter. If not specified, the coupling facility does not spill excess structure objects into storage-class memory for this structure.

SCMALGORITHM

Identifies the algorithm that the coupling facility uses to control the movement of structure objects between coupling facility real storage and storage-class memory in an effort to mitigate the performance penalty associated with its use.

SCMALGORITHM is a required parameter when SCMMAXSIZE is specified and is not allowed for specification otherwise.

ALLOWAUTOALT(NO)**ALLOWAUTOALT(YES)**

Specifies the installation's request to allow system-initiated alters (automatic alter) for this structure. For structure alter processing to be started for a structure, alter must be permitted (see the SETXCF MODIFY command) and the exploiter must also allow alter. The ALLOWAUTOALT specification affects the default value for MINSIZE.

ALLOWAUTOALT is an optional parameter. NO is the default, when ALLOWAUTOALT is not specified.

FULLTHRESHOLD

Specifies a percentage value used by the system to control structure full monitoring and automatic alter (see ALLOWAUTOALT). The value specifies a percent full threshold for the structure. For a cache structure, the percent full is based on changed coupling facility structure objects. For a list or lock structure, the percent full is based on in use coupling facility structure objects. This number is specified as a percentage and can be 1 - 3 decimal digits long (0-100). The number must be greater than or equal to 0 and less than or equal to 100.

FULLTHRESHOLD is an optional parameter. 80% is the default, when FULLTHRESHOLD is not specified.

PREFLIST

Specifies an ordered list of coupling facility names from which the system is to choose when allocating a structure in a coupling facility. If ENFORCEORDER(NO) is specified, the system attempts to allocate the structure in the first coupling facility in the preference list that meets the following allocation criteria regulated by the duplex site preference, and listed in order of relative importance from most important to least important.

PREFLIST is a required parameter.

EXCLLIST

Specifies the list of 1 to 8 coupling facility structure names with which this structure should not share the same coupling facility. The system attempts to honor the exclusion request, but will not fail a request to allocate a structure when all other requirements for structure allocation have been satisfied and the exclusion list cannot be honored. However, if all other attributes are equal, a coupling facility containing only one instance of a duplexed structure from the exclusion list is selected over a coupling facility containing a simplex structure from the exclusion list.

EXCLLIST is an optional parameter.

REBUILDPERCENT

Specifies, as a percent of lost connectivity to a structure, when MVS is to initiate a user-managed rebuild. Use of REBUILDPERCENT requires that all active connections to the structure support user-managed rebuild and that the structure is not being used for XCF signaling.

This number is specified as a nonzero percentage and can be 1 - 3 decimal digits long (1-100).

REBUILDPERCENT is an optional parameter. When REBUILDPERCENT is not specified, the default value is 1%.

DUPLEX

Specifies the installation's request for duplexing rebuild of the structure.

DUPLEX is an optional parameter. DISABLED is the default, when DUPLEX is not specified.

RECPRTY

This statement specifies the priority to be given to the structure for LOSSCONN recovery (system loses connectivity to a coupling facility) and policy-initiated duplexing for DUPLEX(ENABLED) structures. The system might defer processing for a "less important" (higher numeric priority value) structure to prevent the "less important" recovery processing from interfering with recovery processing for a "more important" (lower numeric priority value) structure. Note that LOSSCONN recovery for a "more important" structure might be impacted when that recovery depends on processing for a "less important" structure. For this reason, be extremely careful when assigning a RECPRTY value.

This number is specified as a 1-digit decimal value (1-4).

RECPRTY is an optional parameter. When RECPRTY is not specified, the system takes a default. When the structure name is ISGLOCK, the default is 1. Otherwise, the default is 3.

SUBNOTIFYDELAY

Specifies the number of microseconds for sublist notification delay time (SLND time). This value refers to delay between the time when a single selected shared message queue exploiter instance is notified of sublist transition from the empty to not-empty state and the time when the other instances are notified. It can be that the other instances are never notified depending on the processing done by the initial exploiter.

The value that is specified for SUBNOTIFYDELAY can be 1 to 7 decimal digits in a range of 0 to 1000000 (1 million) microseconds.

SUBNOTIFYDELAY is an optional parameter. The default value when SUBNOTIFYDELAY is not specified is 5000 microseconds.

LISTNOTIFYDELAY

Specifies the number of microseconds for the list notification delay time (LND time). This value refers to the delay between the time when a single registered list monitor is notified of a list state transition from the empty to not-empty state and the time when the other registered list monitor instances of a list are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for LISTNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. LISTNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

LISTNOTIFYDELAY is an optional parameter. The default value when LISTNOTIFYDELAY is not specified is 0 microseconds.

KEYRNOTIFYDELAY

Specifies the number of microseconds for the keyrange notification delay time (KRND time). This value refers to the delay between the time when a single registered keyrange monitor is notified of a keyrange state transition from the empty to not-empty state and the time when the other registered keyrange monitoring instances of a list keyrange are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for KEYRNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. KEYRNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

KEYRNOTIFYDELAY is an optional parameter. The default value when KEYRNOTIFYDELAY is not specified is 0 microseconds.

ENFORCEORDER

When ENFORCEORDER(YES) is specified, allocation attempts adhere to the PREFLIST more closely, giving it a higher priority than such factors as system weight connectivity and optimum size. Use of this option can help ensure failure-isolation between a coupling facility and the systems that utilize it (for example, when these resources reside on the same CEC).

ENFORCEORDER is an optional parameter. ENFORCEORDER(NO) is the default, when ENFORCEORDER is not specified.

ALLOWREALLOCATE

Specifies the installation's request for REALLOCATE processing when evaluation of the allocated structure determines that the structure needs rebuild processing.

ALLOWREALLOCATE is an optional parameter. When ALLOWREALLOCATE is not specified, the default value is YES.

ENCRYPT

Specifies whether list and cache structure entry data and entry adjunct data written to the structure and residing in the structure should be encrypted. The structure entry and entry adjunct

data is in an encrypted format while the data is being transferred to and from the coupling facility and while the data resides in the coupling facility structure. Encrypted data is decrypted when read from the structure.

Example

In the following example, the GET method is used to retrieve a list of the CFRM policy properties.

```
{
  "data type": "CFRM",
  "_version_supported": "4",
  "policy": [
    {
      "name": "CTTEOLD",
      "_defined": "02/25/2020T12:04:56.187608Z",
      "user": "IBMUSER",
      "_version": "0",
      "cf": [
        {
          "name": "TESTCF",
          "type": "00CF01",
          "mfg": "XXX",
          "plant": "XX",
          "sequence": "XXXXXXXXXXXX",
          "partition": "00",
          "cpcid": "00",
          "dumpspace": 2000
        },
        {
          "name": "LF01",
          "type": "00ND01",
          "mfg": "XXX",
          "plant": "XX",
          "sequence": "XXXXXXXXXXXX",
          "partition": "00",
          "cpcid": "00"
        }
      ],
      "structure": [
        {
          "name": "LF01",
          "subnotifydelay": 20,
          "size": 2000,
          "recprty": 3,
          "fullthreshold": 50,
          "preflist": [
            "LF01",
            "TESTCF"
          ]
        }
      ]
    }
  ]
}
```

Figure 287. Sample response from a request to retrieve a list of the CFRM policy properties.

Activate CFRM policy

This operation activates a CFRM policy.

HTTP method and URI path

```
PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/activation/<pol-name>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm

- **pol-name** is the URI path variable that identifies the name of a policy.

Description

On successful completion, the request will activate a specified CFRM policy.

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

None.

Example

In the following request example, the PUT method is used to activate a specified CFRM policy.

```
PUT zosmf/sysplex/rest/v1/policies/cfrm/activation/policy1
```

```
{
  "_status": "completed",
  "_json_version": 4
}
```

Figure 288. Sample request to activate a specified CFRM policy.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

The user’s z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>. ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. If the request is successful, the response also includes a JSON object. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 658.

Rename existing CFRM policy

This operation renames an existing CFRM policy.

HTTP method and URI path

```
PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/name/<pol-name>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm
- **pol-name** is the URI path variable that identifies the name of a policy.

Description

On successful completion, the request will rename an existing CFRM policy to a new name.

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

Your request must include the new name of the CFRM policy.

Example

In the following request example, the PUT method is used to rename an existing CFRM policy to a new name.

```
PUT /zosmf/sysplex/rest/v1/policies/cfrm/name/<old-name>
```

```
{  
  "name": "newName"  
}
```

Figure 289. Sample request to rename an existing CFRM policy to a new name.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

The user’s z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>. ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.
- Update access for MVSADMIN.XCF.CFRM in FACILITY class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 (No content) is returned without a response body and indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 658](#).

Modify CFRM policy properties

This operation modifies the properties of a CFRM policy.

HTTP method and URI path

```
PUT /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm
- **pol-name** is the URI path variable that identifies the name of a policy.

Description

On successful completion, the request will update the CFRM policy's corresponding properties.

For a description of the request content, see [“Request content” on page 670](#).

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

The request content is required, but some properties are optional.

policy

You can modify one policy at a time.

name

Indicates the name of the CFRM policy.

_defined

Indicates the date and time of modification of the CFRM policy.

CF

Specifies the definition of a coupling facility within the scope of the named policy. The location of the information that is needed to define the coupling facility in a CFRM policy depends on the processor on which the coupling facility is installed.

The limit for the number of coupling facilities that can be defined in a policy is established when the CFRM couple data set is formatted.

The coupling facility must appear in the preference list of at least one coupling facility structure that is defined within the policy or the administrative policy utility flags the statement as an error.

NAME

Specifies the 1 - 8 character length name of the coupling facility. The valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), national characters (\$,@,#), or an underscore (_). The **cfname** must start with an alphabetic character (A-Z).

Consider defining your coupling facility name to match the name of the LPAR in which the coupling facility is to run.

NAME is a required parameter.

The following set of parameters describe the unique coupling facility that is defined in the policy.

TYPE

Specifies the 6-character machine type. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. TYPE is a required parameter.

MFG

Specifies the 3-character manufacturer identification. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). MFG is a required parameter.

PLANT

Specifies the 2-character plant of manufacture code. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). PLANT is a required parameter.

SEQUENCE

Specifies the 12-character serial number. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. SEQUENCE is a required parameter.

PARTITION

Specifies the 1- or 2-hexadecimal digit qualifier to uniquely identify and associate a coupling facility to a specific PR/SM partition.

- If the coupling facility is defined on a System z9®, z990, and z890 processor, then the PARTITION value must match the partition ID associated on the activation profile for the CF image on the support element or hardware master console.
- If the coupling facility is defined on a pre-z990 or pre-z890 or earlier processor, then the PARTITION value must match the partition number specified by HCD/IOCP in the IOCDs for that partition and can be in the range of 01-0F.
- Otherwise, the PARTITION value can be in the range of 0-F and is the same as the partition number defined in HCD.

Partition is a required parameter.

SIDE

Specifies the 1-numeric character to identify the coupling facility to a specific physical side of a CPC running in Physically Partitioned mode. You should not split or merge physical sides on which a coupling facility resides.

Use caution when splitting or merging physical sides of a processor that contains a coupling facility.

SIDE is an optional parameter. Its omission implies that the coupling facility resides on a single image CPC.

- SIDE(0) — Side 0 of a physically partitioned machine. Do not specify for a Single Image CPC.
- SIDE(1) — Side 1 of a physically partitioned machine. Do not specify for a Single Image CPC.

CPCID

Specifies the 2-hexadecimal digit qualifier (00-FF) to identify a coupling facility to a specific CPC in a CMOS processor. CPCID is an optional parameter.

DUMPSPACE

Specifies the amount of space to be reserved in the coupling facility for dumping structures allocated in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1-10 decimal digits long. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

DUMPSPACE is an optional parameter. If omitted, no storage in the coupling facility is reserved for dumping.

SITE

Specifies the site at which the coupling facility resides. When specified, this information (along with status information from a Recovery Manager (for example, GDPS®)), will be used in break duplexing decisions to keep the structure instance at the recovery site. SITE changes take effect when the policy is activated.

This keyword should be considered when specifying the DUPLEX **dupsite** and **dupmode** parameters as part of the STRUCTURE definition.

STRUCTURE

Specifies the definition of a structure within the scope of the named policy. The limit for the number of structures that can be defined in a policy is established when the CFRM couple data set is formatted.

NAME

Specifies the 1-16 character name of the structure. The valid characters are numeric characters, uppercase alphabetic characters, national characters (\$,@,#), or an underscore (_). The *strname* must start with an alphabetic character (A-Z). IBM names begin with SYS, or the letters A through I.

Structures that are used for XCF signaling must begin with the letters IXC.

NAME is a required parameter.

SIZE

Specifies the maximum amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long.

SIZE is a required parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

INITSIZE

Specifies the initial amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number must be 1 - 10 decimal digits long. The INITSIZE value must be less than or equal to the SIZE value. Otherwise, the system issues error message IXC745I.

INITSIZE is an optional parameter. If not specified, the system uses the SIZE parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

MINSIZE

Specifies the smallest size to which the structure can ever be altered, preventing allocation of an unusable structure. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long. The MINSIZE value must be less than or equal to the INITSIZE value, or less than or equal to the SIZE value if INITSIZE is not specified.

MINSIZE is an optional parameter. The largest size that can be specified is 1T. Sizes larger than 1T will cause message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value. If not specified and ALLOWAUTOALT(NO) is specified or defaulted to, the MINSIZE used will be 0. If not specified and ALLOWAUTOALT(YES) is specified, the system will set MINSIZE to 75% of INITSIZE, or to 75% of SIZE if INITSIZE is not specified.

SCMMAXSIZE

Specifies the maximum amount of storage-class (“flash”) memory that is assignable for use by this structure in the coupling facility. The number is specified in an integer without any unit.

K (kilobytes) is the default unit. Valid values for the number are 1 - 11 decimal digits long and greater than 0, if this keyword is specified.

SCMMAXSIZE is an optional parameter. If not specified, the coupling facility does not spill excess structure objects into storage-class memory for this structure.

SCMALGORITHM

Identifies the algorithm that the coupling facility uses to control the movement of structure objects between coupling facility real storage and storage-class memory in an effort to mitigate the performance penalty associated with its use.

SCMALGORITHM is a required parameter when SCMMAXSIZE is specified and is not allowed for specification otherwise.

ALLOWAUTOALT(NO)

ALLOWAUTOALT(YES)

Specifies the installation's request to allow system-initiated alters (automatic alter) for this structure. For structure alter processing to be started for a structure, alter must be permitted (see the SETXCF MODIFY command) and the exploiter must also allow alter. The ALLOWAUTOALT specification affects the default value for MINSIZE.

ALLOWAUTOALT is an optional parameter. NO is the default, when ALLOWAUTOALT is not specified.

FULLTHRESHOLD

Specifies a percentage value used by the system to control structure full monitoring and automatic alter (see ALLOWAUTOALT). The value specifies a percent full threshold for the structure. For a cache structure, the percent full is based on changed coupling facility structure objects. For a list or lock structure, the percent full is based on in use coupling facility structure objects. This number is specified as a percentage and can be 1 - 3 decimal digits long (0-100). The number must be greater than or equal to 0 and less than or equal to 100.

FULLTHRESHOLD is an optional parameter. 80% is the default, when FULLTHRESHOLD is not specified.

PREFLIST

Specifies an ordered list of coupling facility names from which the system is to choose when allocating a structure in a coupling facility. If ENFORCEORDER(NO) is specified, the system attempts to allocate the structure in the first coupling facility in the preference list that meets the following allocation criteria regulated by the duplex site preference, and listed in order of relative importance from most important to least important.

PREFLIST is a required parameter.

EXCLLIST

Specifies the list of 1 to 8 coupling facility structure names with which this structure should not share the same coupling facility. The system attempts to honor the exclusion request, but will not fail a request to allocate a structure when all other requirements for structure allocation have been satisfied and the exclusion list cannot be honored. However, if all other attributes are equal, a coupling facility containing only one instance of a duplexed structure from the exclusion list is selected over a coupling facility containing a simplex structure from the exclusion list.

EXCLLIST is an optional parameter.

REBUILDPERCENT

Specifies, as a percent of lost connectivity to a structure, when MVS is to initiate a user-managed rebuild. Use of REBUILDPERCENT requires that all active connections to the structure support user-managed rebuild and that the structure is not being used for XCF signaling.

This number is specified as a nonzero percentage and can be 1 - 3 decimal digits long (1-100).

REBUILDPERCENT is an optional parameter. When REBUILDPERCENT is not specified, the default value is 1%.

DUPLEX

Specifies the installation's request for duplexing rebuild of the structure.

DUPLEX is an optional parameter. DISABLED is the default, when DUPLEX is not specified.

RECPRTY

This statement specifies the priority to be given to the structure for LOSSCONN recovery (system loses connectivity to a coupling facility) and policy-initiated duplexing for DUPLEX(ENABLED) structures. The system might defer processing for a "less important" (higher numeric priority value) structure to prevent the "less important" recovery processing from interfering with recovery processing for a "more important" (lower numeric priority value) structure. Note that LOSSCONN recovery for a "more important" structure might be impacted when that recovery depends on processing for a "less important" structure. For this reason, be extremely careful when assigning a RECPRTY value.

This number is specified as a 1-digit decimal value (1-4).

RECPRTY is an optional parameter. When RECPRTY is not specified, the system takes a default. When the structure name is ISGLOCK, the default is 1. Otherwise, the default is 3.

SUBNOTIFYDELAY

Specifies the number of microseconds for sublist notification delay time (SLND time). This value refers to delay between the time when a single selected shared message queue exploiter instance is notified of sublist transition from the empty to not-empty state and the time when the other instances are notified. It can be that the other instances are never notified depending on the processing done by the initial exploiter.

The value that is specified for SUBNOTIFYDELAY can be 1 to 7 decimal digits in a range of 0 to 1000000 (1 million) microseconds.

SUBNOTIFYDELAY is an optional parameter. The default value when SUBNOTIFYDELAY is not specified is 5000 microseconds.

LISTNOTIFYDELAY

Specifies the number of microseconds for the list notification delay time (LND time). This value refers to the delay between the time when a single registered list monitor is notified of a list state transition from the empty to not-empty state and the time when the other registered list monitor instances of a list are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for LISTNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. LISTNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

LISTNOTIFYDELAY is an optional parameter. The default value when LISTNOTIFYDELAY is not specified is 0 microseconds.

KEYRNOTIFYDELAY

Specifies the number of microseconds for the keyrange notification delay time (KRND time). This value refers to the delay between the time when a single registered keyrange monitor is notified of a keyrange state transition from the empty to not-empty state and the time when the other registered keyrange monitoring instances of a list keyrange are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for KEYRNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. KEYRNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

KEYRNOTIFYDELAY is an optional parameter. The default value when KEYRNOTIFYDELAY is not specified is 0 microseconds.

ENFORCEORDER

When ENFORCEORDER(YES) is specified, allocation attempts adhere to the PREFLIST more closely, giving it a higher priority than such factors as system weight connectivity and optimum size. Use of this option can help ensure failure-isolation between a coupling facility and the systems that utilize it (for example, when these resources reside on the same CEC).

ENFORCEORDER is an optional parameter. ENFORCEORDER(NO) is the default, when ENFORCEORDER is not specified.

ALLOWREALLOCATE

Specifies the installation's request for REALLOCATE processing when evaluation of the allocated structure determines that the structure needs rebuild processing.

ALLOWREALLOCATE is an optional parameter. When ALLOWREALLOCATE is not specified, the default value is YES.

ENCRYPT

Specifies whether list and cache structure entry data and entry adjunct data written to the structure and residing in the structure should be encrypted. The structure entry and entry adjunct data is in an encrypted format while the data is being transferred to and from the coupling facility and while the data resides in the coupling facility structure. Encrypted data is decrypted when read from the structure.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

The user's z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>.ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.
- Update access for MVSADMIN.XCF.CFRM in FACILITY class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 (No content) is returned without a response body and indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 658](#).

Example

In the following example, the PUT method is used to modify the properties of a CFRM policy.

```
PUT /zosmf/sysplex/rest/v1/policies/cfrm/<pol-name>
```

```
{
  "policy":[
    {
      "name":"CTTEOLD",
      "_defined":"02/25/2020T12:04:56.187608Z",
      "cf":[
        {
          "name":"TESTCF",
          "type":"00CF01",
          "mfg":"XXX",
          "plant":"XX",
          "sequence":"XXXXXXXXXXXX",
          "partition":"00",
          "cpcid":"00",
          "dumpspace": 2000
        },
        {
          "name":"LF01",
          "type":"00ND01",
          "mfg":"XXX",
          "plant":"XX",
          "sequence":"XXXXXXXXXXXX",
          "partition":"00",
          "cpcid":"00"
        }
      ],
      "structure":[
        {
          "name":"LF01",
          "subnotifydelay":20,
          "size":2000,
          "recprty":3,
          "fullthreshold":50,
          "preflist":[
            "LF01",
            "TESTCF"
          ]
        }
      ]
    }
  ]
}
```

Figure 290. Sample request to modify the properties of a CFRM policy.

Add CFRM policies to administrative policy

This operation adds a single policy or multiple CFRM policies to an administrative policy.

HTTP method and URI path

```
POST /zosmf/sysplex/rest/<version>/policies/<policy-type>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm

Description

On successful completion, the request will add a single policy or multiple CFRM policies to an administrative policy

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

The request content is required, but some properties are optional.

policy

You can add a single policy or multiple CFRM policies to an administrative policy.

name

Indicates the name of the CFRM policy.

CF

Specifies the definition of a coupling facility within the scope of the named policy. The location of the information that is needed to define the coupling facility in a CFRM policy depends on the processor on which the coupling facility is installed.

The limit for the number of coupling facilities that can be defined in a policy is established when the CFRM couple data set is formatted.

The coupling facility must appear in the preference list of at least one coupling facility structure that is defined within the policy or the administrative policy utility flags the statement as an error.

NAME

Specifies the 1 - 8 character length name of the coupling facility. The valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), national characters (\$,@,#), or an underscore (_). The **cfname** must start with an alphabetic character (A-Z).

Consider defining your coupling facility name to match the name of the LPAR in which the coupling facility is to run.

NAME is a required parameter.

The following set of parameters describe the unique coupling facility that is defined in the policy.

TYPE

Specifies the 6-character machine type. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. TYPE is a required parameter.

MFG

Specifies the 3-character manufacturer identification. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). MFG is a required parameter.

PLANT

Specifies the 2-character plant of manufacture code. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9). PLANT is a required parameter.

SEQUENCE

Specifies the 12-character serial number. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary. SEQUENCE is a required parameter.

PARTITION

Specifies the 1- or 2-hexadecimal digit qualifier to uniquely identify and associate a coupling facility to a specific PR/SM partition.

- If the coupling facility is defined on a System z9®, z990, and z890 processor, then the PARTITION value must match the partition ID associated on the activation profile for the CF image on the support element or hardware master console.
- If the coupling facility is defined on a pre-z990 or pre-z890 or earlier processor, then the PARTITION value must match the partition number specified by HCD/IOCP in the IOCDS for that partition and can be in the range of 01-0F.
- Otherwise, the PARTITION value can be in the range of 0-F and is the same as the partition number defined in HCD.

Partition is a required parameter.

SIDE

Specifies the 1-numeric character to identify the coupling facility to a specific physical side of a CPC running in Physically Partitioned mode. You should not split or merge physical sides on which a coupling facility resides.

Use caution when splitting or merging physical sides of a processor that contains a coupling facility.

SIDE is an optional parameter. Its omission implies that the coupling facility resides on a single image CPC.

- SIDE(0) — Side 0 of a physically partitioned machine. Do not specify for a Single Image CPC.
- SIDE(1) — Side 1 of a physically partitioned machine. Do not specify for a Single Image CPC.

CPCID

Specifies the 2-hexadecimal digit qualifier (00-FF) to identify a coupling facility to a specific CPC in a CMOS processor. CPCID is an optional parameter.

DUMPSPACE

Specifies the amount of space to be reserved in the coupling facility for dumping structures allocated in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1-10 decimal digits long. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

DUMPSPACE is an optional parameter. If omitted, no storage in the coupling facility is reserved for dumping.

SITE

Specifies the site at which the coupling facility resides. When specified, this information (along with status information from a Recovery Manager (for example, GDPS®)), will be used in break duplexing decisions to keep the structure instance at the recovery site. SITE changes take effect when the policy is activated.

This keyword should be considered when specifying the DUPLEX **dupsite** and **dupmode** parameters as part of the STRUCTURE definition.

STRUCTURE

Specifies the definition of a structure within the scope of the named policy. The limit for the number of structures that can be defined in a policy is established when the CFRM couple data set is formatted.

NAME

Specifies the 1-16 character name of the structure. The valid characters are numeric characters, uppercase alphabetic characters, national characters (\$,@,#), or an underscore (_). The *strname* must start with an alphabetic character (A-Z). IBM names begin with SYS, or the letters A through I.

Structures that are used for XCF signaling must begin with the letters IXC.

NAME is a required parameter.

SIZE

Specifies the maximum amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long.

SIZE is a required parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

INITSIZE

Specifies the initial amount of space to be allocated for the structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number must be 1 - 10 decimal digits long. The INITSIZE value must be less than or equal to the SIZE value. Otherwise, the system issues error message IXC745I.

INITSIZE is an optional parameter. If not specified, the system uses the SIZE parameter. The largest size that can be specified is 1T. Sizes larger than 1T causes message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value.

MINSIZE

Specifies the smallest size to which the structure can ever be altered, preventing allocation of an unusable structure. The number is specified in an integer without any unit. K (kilobytes) is the default unit. The number can be 1 - 10 decimal digits long. The MINSIZE value must be less than or equal to the INITSIZE value, or less than or equal to the SIZE value if INITSIZE is not specified.

MINSIZE is an optional parameter. The largest size that can be specified is 1T. Sizes larger than 1T will cause message IXC730I to be displayed, indicating that the number specified is not within the allowable range for the keyword value. If not specified and ALLOWAUTOALT(NO) is specified or defaulted to, the MINSIZE used will be 0. If not specified and ALLOWAUTOALT(YES) is specified, the system will set MINSIZE to 75% of INITSIZE, or to 75% of SIZE if INITSIZE is not specified.

SCMMAXSIZE

Specifies the maximum amount of storage-class (“flash”) memory that is assignable for use by this structure in the coupling facility. The number is specified in an integer without any unit. K (kilobytes) is the default unit. Valid values for the number are 1 - 11 decimal digits long and greater than 0, if this keyword is specified.

SCMMAXSIZE is an optional parameter. If not specified, the coupling facility does not spill excess structure objects into storage-class memory for this structure.

SCMALGORITHM

Identifies the algorithm that the coupling facility uses to control the movement of structure objects between coupling facility real storage and storage-class memory in an effort to mitigate the performance penalty associated with its use.

SCMALGORITHM is a required parameter when SCMMAXSIZE is specified and is not allowed for specification otherwise.

ALLOWAUTOALT(NO)**ALLOWAUTOALT(YES)**

Specifies the installation's request to allow system-initiated alters (automatic alter) for this structure. For structure alter processing to be started for a structure, alter must be permitted (see the SETXCF MODIFY command) and the exploiter must also allow alter. The ALLOWAUTOALT specification affects the default value for MINSIZE.

ALLOWAUTOALT is an optional parameter. NO is the default, when ALLOWAUTOALT is not specified.

FULLTHRESHOLD

Specifies a percentage value used by the system to control structure full monitoring and automatic alter (see ALLOWAUTOALT). The value specifies a percent full threshold for the

structure. For a cache structure, the percent full is based on changed coupling facility structure objects. For a list or lock structure, the percent full is based on in use coupling facility structure objects. This number is specified as a percentage and can be 1 - 3 decimal digits long (0-100). The number must be greater than or equal to 0 and less than or equal to 100.

FULLTHRESHOLD is an optional parameter. 80% is the default, when FULLTHRESHOLD is not specified.

PREFLIST

Specifies an ordered list of coupling facility names from which the system is to choose when allocating a structure in a coupling facility. If ENFORCEORDER(NO) is specified, the system attempts to allocate the structure in the first coupling facility in the preference list that meets the following allocation criteria regulated by the duplex site preference, and listed in order of relative importance from most important to least important.

PREFLIST is a required parameter.

EXCLLIST

Specifies the list of 1 to 8 coupling facility structure names with which this structure should not share the same coupling facility. The system attempts to honor the exclusion request, but will not fail a request to allocate a structure when all other requirements for structure allocation have been satisfied and the exclusion list cannot be honored. However, if all other attributes are equal, a coupling facility containing only one instance of a duplexed structure from the exclusion list is selected over a coupling facility containing a simplex structure from the exclusion list.

EXCLLIST is an optional parameter.

REBUILDPERCENT

Specifies, as a percent of lost connectivity to a structure, when MVS is to initiate a user-managed rebuild. Use of REBUILDPERCENT requires that all active connections to the structure support user-managed rebuild and that the structure is not being used for XCF signaling.

This number is specified as a nonzero percentage and can be 1 - 3 decimal digits long (1-100).

REBUILDPERCENT is an optional parameter. When REBUILDPERCENT is not specified, the default value is 1%.

DUPLEX

Specifies the installation's request for duplexing rebuild of the structure.

DUPLEX is an optional parameter. DISABLED is the default, when DUPLEX is not specified.

RECPRTY

This statement specifies the priority to be given to the structure for LOSSCONN recovery (system loses connectivity to a coupling facility) and policy-initiated duplexing for DUPLEX(ENABLED) structures. The system might defer processing for a "less important" (higher numeric priority value) structure to prevent the "less important" recovery processing from interfering with recovery processing for a "more important" (lower numeric priority value) structure. Note that LOSSCONN recovery for a "more important" structure might be impacted when that recovery depends on processing for a "less important" structure. For this reason, be extremely careful when assigning a RECPRTY value.

This number is specified as a 1-digit decimal value (1-4).

RECPRTY is an optional parameter. When RECPRTY is not specified, the system takes a default. When the structure name is ISGLOCK, the default is 1. Otherwise, the default is 3.

SUBNOTIFYDELAY

Specifies the number of microseconds for sublist notification delay time (SLND time). This value refers to delay between the time when a single selected shared message queue exploiter instance is notified of sublist transition from the empty to not-empty state and

the time when the other instances are notified. It can be that the other instances are never notified depending on the processing done by the initial exploiter.

The value that is specified for SUBNOTIFYDELAY can be 1 to 7 decimal digits in a range of 0 to 1000000 (1 million) microseconds.

SUBNOTIFYDELAY is an optional parameter. The default value when SUBNOTIFYDELAY is not specified is 5000 microseconds.

LISTNOTIFYDELAY

Specifies the number of microseconds for the list notification delay time (LND time). This value refers to the delay between the time when a single registered list monitor is notified of a list state transition from the empty to not-empty state and the time when the other registered list monitor instances of a list are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for LISTNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. LISTNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

LISTNOTIFYDELAY is an optional parameter. The default value when LISTNOTIFYDELAY is not specified is 0 microseconds.

KEYRNOTIFYDELAY

Specifies the number of microseconds for the keyrange notification delay time (KRND time). This value refers to the delay between the time when a single registered keyrange monitor is notified of a keyrange state transition from the empty to not-empty state and the time when the other registered keyrange monitoring instances of a list keyrange are notified. It can be that the other instances are never notified, depending on the processing done by the initial notified monitor.

The value that is specified for KEYRNOTIFYDELAY can be 1 to 7 decimal digits, in a range of 0 to 1000000 (1 million) microseconds. KEYRNOTIFYDELAY only applies to structure meeting certain criteria and is ignored for all other structures.

KEYRNOTIFYDELAY is an optional parameter. The default value when KEYRNOTIFYDELAY is not specified is 0 microseconds.

ENFORCEORDER

When ENFORCEORDER(YES) is specified, allocation attempts adhere to the PREFLIST more closely, giving it a higher priority than such factors as system weight connectivity and optimum size. Use of this option can help ensure failure-isolation between a coupling facility and the systems that utilize it (for example, when these resources reside on the same CEC).

ENFORCEORDER is an optional parameter. ENFORCEORDER(NO) is the default, when ENFORCEORDER is not specified.

ALLOWREALLOCATE

Specifies the installation's request for REALLOCATE processing when evaluation of the allocated structure determines that the structure needs rebuild processing.

ALLOWREALLOCATE is an optional parameter. When ALLOWREALLOCATE is not specified, the default value is YES.

ENCRYPT

Specifies whether list and cache structure entry data and entry adjunct data written to the structure and residing in the structure should be encrypted. The structure entry and entry adjunct data is in an encrypted format while the data is being transferred to and from the coupling facility and while the data resides in the coupling facility structure. Encrypted data is decrypted when read from the structure.

Example

In the following request example, the POST method is used to add a CFRM policy to an administrative policy.

POST /zosmf/sysplex/rest/v1/policies/cfrm

```
{
  "policy": [
    {
      "cf": [
        {
          "dumpspace": 256,
          "sequence": "XXXXXXXXXX",
          "partition": "00",
          "cpcid": "00",
          "plant": "XX",
          "name": "TESTCF",
          "type": "00CF01",
          "mfg": "XXX"
        },
        {
          "dumpspace": 4,
          "sequence": "XXXXXXXXXX",
          "partition": "00",
          "cpcid": "00",
          "plant": "XX",
          "name": "LF01",
          "type": "00ND01",
          "mfg": "XXX"
        }
      ],
      "name": "CTTENEH",
      "structure": [
        {
          "preflist": [
            "LF01",
            "TESTCF"
          ],
          "size": 1000,
          "name": "LT02"
        }
      ]
    }
  ],
}
```

```
    "cf": [
      {
        "dumpspace": 256,
        "sequence": "XXXXXXXXXX",
        "partition": "00",
        "cpcid": "00",
        "plant": "XX",
        "name": "TESTCF",
        "type": "00CF01",
        "mfg": "XXX"
      },
      {
        "dumpspace": 4,
        "sequence": "XXXXXXXXXX",
        "partition": "00",
        "cpcid": "00",
        "plant": "XX",
        "name": "LF01",
        "type": "00ND01",
        "mfg": "XXX"
      }
    ],
    "name": "CTTENEW",
    "structure": [
      {
        "preflist": [
          "LF01",
          "TESTCF"
        ],
        "size": 1000,
        "name": "LT02"
      }
    ]
  }
]
```

Figure 291. Sample request to add a CFRM policy to an administrative policy.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

The user’s z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>. ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.
- Update access for MVSADMIN.XCF.CFRM in FACILITY class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 (Created) is returned without a response body and indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 658.

Copy existing CFRM policy with a new name

This operation adds a new policy by copying an existing policy with a new name.

HTTP method and URI path

```
POST /zosmf/sysplex/rest/<version>/policies/<policy-type>/<pol-name>
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm
- **pol-name** further qualifies the request and indicates the specific name of the policy.

Description

On successful completion, the request will add a new policy by copying an existing policy with a new name.

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

Your request must include the name of the new CFRM policy, which is copied from an existing CFRM policy.

Example

In the following request example, the POST method is used to copy an existing policy with a new name.

```
POST /zosmf/sysplex/rest/v1/policies/cfrm/CTTEST1
```

```
{  
  "name": "CTTEST2"  
}
```

Figure 292. Sample request to copy an existing policy with a new name

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

The user’s z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>. ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.
- Update access for MVSADMIN.XCF.CFRM in FACILITY class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 (Created) is returned without a response body and indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 658.

Delete multiple CFRM policies from an administrative policy

This operation deletes a single policy or multiple CFRM policies from an administrative policy.

HTTP method and URI path

```
DELETE /zosmf/sysplex/rest/<version>/policies/cfrm?name=<pol-name>&name=<pol-name2>...
```

where:

- **version** is the URI path variable that identifies the version of the z/OS sysplex management service. The following value is valid: v1
- **policy-type** is the URI path variable that identifies a policy type. The following value is valid: cfrm
- **pol-name** further qualifies the request and indicates the specific names of the policies to be deleted.

Description

On successful completion, the request will delete a single policy or multiple CFRM policies from an administrative policy.

Standard headers

Use the following standard HTTP header with this request: Content-Type: application/json

Custom headers

None.

Request content

Your request must include the names of the CFRM policies to be deleted.

Examples

In the following request example, the DELETE method is used to delete a CFRM policy from an administrative policy.

```
DELETE /zosmf/sysplex/rest/v1/policies/cfrm?name=CTTEST1
```

Figure 293. Sample request to delete a CFRM policy from an administrative policy.

In the following request example, the DELETE method is used to delete multiple CFRM policies from an administrative policy.

```
DELETE /zosmf/sysplex/rest/v1/policies/cfrm?name=CTTEST1&name=ALTERPOL
```

Figure 294. Sample request to delete multiple CFRM policies from an administrative policy.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

The user’s z/OS user ID must have access to the following resource profiles:

- READ access for <SAF-prefix>. ZOSMF.SYSPLEX.MODIFY in the ZMFAPLA class.
- Update access for MVSADMIN.XCF.CFRM in FACILITY class.

Response content

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 (No content) is returned without a response body and indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 658.

Error report

This section describes the possible errors that can be returned in the JSON error report document for sysplex management API services requests.

Error reason codes

[Table 372 on page 686](#) shows the possible error reason codes and messages.

Table 372. Errors for sysplex management API services operations	
Reason Code	Message
1	User is not authorized to view the CFRM policy.
3	User is not authorized to modify the CFRM policy.
4	User is not authorized to use the Sysplex Management.
5	The input value contains a syntax error.

Table 372. Errors for sysplex management API services operations (continued)

Reason Code	Message
6	The policy input json is invalid, only support one policy at a time.
7	Can't get an invalid policy from system.
9	The input json includes an invalid key.
11	The required attribute is missing.
13	The input json includes an invalid value.
15	The policy's value type is incorrect.
16	The input json is invalid.
17	The required policy name is missing in the URI.
19	Convert policy json string to policy object failed.
21	Convert policy object to policy json string failed.
23	Json and object transition error.
25	The policy specified is not existed.
26	The specified policy is out of date, retrieve the latest content and retry the request.
27	Parse the json failed when convert the json to string.
29	The request couldn't be completed because an error occurred when call the IXCAPU.
37	Copy the policy error, retrieve the policy failed.
39	The request could not be completed because an error occurred.
45	User is not authorized to activate the policy.
47	Update the persistence file failed when active the policy.
49	Active the policy failed.
51	Read the CFRM policy data failed.

Topology services

The topology services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with the groups, sysplexes, central processor complexes (CPCs), and systems that are defined to z/OSMF, as described in this topic.

Table 373 on page 687 lists the operations that the topology services provide.

Table 373. Operations provided through the topology services.

Operation	HTTP method and URI path
“List the systems defined to z/OSMF” on page 689	GET /zosmf/resttopology/systems

Table 373. Operations provided through the topology services. (continued)

Operation	HTTP method and URI path
“List the groups defined to z/OSMF” on page 692	GET /zosmf/resttopology/groups
“List the systems included in a group” on page 694	GET /zosmf/resttopology/systems/groupName/<groupName>
“List the sysplexes defined to z/OSMF” on page 697	GET /zosmf/resttopology/sysplexes
“List the systems included in a sysplex” on page 698	GET /zosmf/resttopology/systems/sysplexName/<sysplexName>
“List the systems included in a CPC” on page 701	GET /zosmf/resttopology/systems/cpcName/<cpcName>
“Validate the status of the selected system” on page 703	GET /zosmf/services/systems/v1/validation/system
“Check all MVS systems in the local plex” on page 709	GET /zosmf/services/systems/v1/validation/plex

Required authorizations

To submit requests through the topology services, your user ID requires authorization to the Systems task provided in z/OSMF. Ensure that your user ID has READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.SETTINGS.SYSTEMS.VIEW. By default, users with user IDs connected to the IZUADMIN and IZUUSER security groups can access the topology services.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data. The following JSON object is used by all topology services for returning data about the requested operation:

```
{
  "items": "item-list",
  "numRows": "total-items"
}
```

where:

item-list

Array that contains the items that were retrieved. The attributes provided in the array depend on the requested operation.

total-items

Number of items retrieved.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 401 Unauthorized

The submitter of the request did not authenticate to z/OSMF or is not authorized to use the topology services.

HTTP 403 Forbidden

The server rejected the request.

HTTP 404 Bad URL

The target of the request (a URL) was not found.

HTTP 405 Method not allowed

The service does not support the HTTP method specified for the request.

HTTP 500 Internal server error

A programming error occurred.

Error logging

Errors from the topology services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

List the systems defined to z/OSMF

You can use this operation to obtain a list of the systems that are defined to a z/OSMF instance.

HTTP method and URI path

```
GET /zosmf/resttopology/systems
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems** informs the service that the request is to retrieve a list of the systems that are defined to the z/OSMF instance.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 688](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 689](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

system-nickname

Unique name assigned to the system definition.

system-name

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

sysplex-name

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

group-names

Comma-separated list of the groups to which the system is assigned.

url

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

zos-level

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where V stands for version, xx is the version number, R stands for release, and yy is the release number. For example, z/OS V2R1.

JES-member-name

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

JES-type

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

CPC-name

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

CPC-serial

Serial number of the CPC.

proxy-name

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

server-name

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

total-items

Number of system definitions that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/systems HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 295. Sample request to retrieve a list of systems

A sample response is shown in [Figure 296 on page 692](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys2",
      "systemName": "sys2",
      "sysplexName": "plex2",
      "groupNames": "production",
      "url": "https://zosmf2.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY2",
      "jesType": "JES3",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs-sftp"
    },
    {
      "systemNickName": "sys3",
      "systemName": "sys3",
      "sysplexName": "plex3",
      "groupNames": "test",
      "url": "https://zosmf3.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY3",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "3"
}

```

Figure 296. Sample response from a request to retrieve a list of systems

List the groups defined to z/OSMF

You can use this operation to obtain a list of the groups that are defined to a z/OSMF instance.

HTTP method and URI path

```
GET /zosmf/resttopology/groups
```

where:

- **zosmf/resttopology** identifies the topology services.
- **groups** informs the service that the request is to retrieve a list of the groups that are defined to the z/OSMF instance.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 688](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 689](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "name": "group-name",
      "description": "group-description",
      "systemNickNames": "system-nicknames"
    }
  ],
  "numRows": "total-items"
}
```

where:

group-name

Name of the groups defined to z/OSMF. A value of <None> indicates that one or more systems are not assigned to a group.

group-description

Description of the group.

system-nicknames

Comma-separated list of the systems assigned to the group. Each system is identified by its nickname.

total-items

Number of groups that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the groups that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/groups HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 297. Sample request to retrieve a list of groups

A sample response is shown in [Figure 298 on page 694](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "name": "development",
      "description": "This group contains the systems used by development.",
      "systemNickNames": "sys1"
    },
    {
      "name": "production",
      "description": "This group contains the systems that are in production.",
      "systemNickNames": "sys2"
    },
    {
      "name": "test",
      "description": "This group contains the systems that are used for testing code.",
      "systemNickNames": "sys1,sys3"
    }
  ],
  "numRows": "3"
}

```

Figure 298. Sample response from a request to retrieve a list of groups

List the systems included in a group

You can use this operation to obtain a list of the systems that are included in a group.

HTTP method and URI path

```
GET /zosmf/resttopology/systems/groupName/<groupName>
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/groupName** informs the service that the request is to retrieve a list of the systems that are defined to a specific group.
- **<groupName>** identifies the group for which to obtain the list of systems. If the group name contains a number sign (#), encode the number sign as %23. For example, if the group name is *test#systems*, specify *test%23systems*. Otherwise, the service will truncate *#systems*, and use *test* as the group name.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations” on page 688](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 689](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

system-nickname

Unique name assigned to the system definition.

system-name

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

sysplex-name

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

group-names

Comma-separated list of the groups to which the system is assigned.

url

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

zos-level

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where V stands for version, xx is the version number, R stands for release, and yy is the release number. For example, z/OS V2R1.

JES-member-name

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

JES-type

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

CPC-name

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

CPC-serial

Serial number of the CPC.

proxy-name

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

server-name

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

total-items

Number of system definitions that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are assigned to the group *test*.

```
GET /zosmf/resttopology/systems/groupName/test HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 299. Sample request to retrieve a list of systems included in a group

A sample response is shown in [Figure 300 on page 696](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys3",
      "systemName": "sys3",
      "sysplexName": "plex3",
      "groupNames": "test",
      "url": "https://zosmf3.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY3",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "2"
}
```

Figure 300. Sample response from a request to retrieve a list of systems included in a group

List the sysplexes defined to z/OSMF

You can use this operation to obtain a list of the sysplexes that are defined to a z/OSMF instance.

HTTP method and URI path

```
GET /zosmf/resttopology/sysplexes
```

where:

- **zosmf/resttopology** identifies the topology services.
- **sysplexes** informs the service that the request is to retrieve a list of the sysplexes that are defined to the z/OSMF instance.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 688](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 689](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "sysplexName": "sysplex-name",
      "systemNickNames": "system-nicknames"
    }
  ],
  "numRows": "total-items"
}
```

where:

sysplex-name

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility. A value of *<Not Specified>* indicates that one or more systems are not assigned to a sysplex.

system-nicknames

Comma-separated list of the systems assigned to the sysplex. Each system is identified by its nickname.

total-items

Number of sysplexes that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the sysplexes that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/sysplexes HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 301. Sample request to retrieve a list of sysplexes

A sample response is shown in [Figure 302 on page 698](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "sysplexName": "plex1",
      "systemNickNames": "sys1"
    },
    {
      "sysplexName": "plex2",
      "systemNickNames": "sys2"
    },
    {
      "sysplexName": "plex3",
      "systemNickNames": "sys3"
    }
  ],
  "numRows": "3"
}
```

Figure 302. Sample response from a request to retrieve a list of sysplexes

List the systems included in a sysplex

You can use this operation to obtain a list of the systems that are included in a sysplex.

HTTP method and URI path

```
GET /zosmf/resttopology/systems/sysplexName/<sysplexName>
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/sysplexName** informs the service that the request is to retrieve a list of the systems that are included in a specific sysplex.
- **<sysplexName>** identifies the sysplex for which to obtain the list of systems.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 688.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 689.

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

system-nickname

Unique name assigned to the system definition.

system-name

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

sysplex-name

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

group-names

Comma-separated list of the groups to which the system is assigned.

url

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

zos-level

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where *V* stands for version, *xx* is the version number, *R* stands for release, and *yy* is the release number. For example, z/OS V2R1.

JES-member-name

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

JES-type

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

CPC-name

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

CPC-serial

Serial number of the CPC.

proxy-name

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

server-name

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

total-items

Number of system definitions that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are included in sysplex *plex1*.

```
GET /zosmf/resttopology/systems/sysplexName/plex1 HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 303. Sample request to retrieve a list of systems included in a sysplex

A sample response is shown in [Figure 304 on page 700](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "1"
}
```

Figure 304. Sample response from a request to retrieve a list of systems included in a sysplex

List the systems included in a CPC

You can use this operation to obtain a list of the systems that are included in a central processor complex (CPC).

HTTP method and URI path

```
GET /zosmf/resttopology/systems/cpcName/<cpcName>
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/cpcName** informs the service that the request is to retrieve a list of the systems that are included in a specific CPC.
- **<cpcName>** identifies the CPC for which to obtain the list of systems.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 688.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 689.

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ]
}
```

```
  },  
  "numRows": "total-items"  
}
```

where:

system-nickname

Unique name assigned to the system definition.

system-name

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

sysplex-name

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

group-names

Comma-separated list of the groups to which the system is assigned.

url

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

zos-level

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where *V* stands for version, *xx* is the version number, *R* stands for release, and *yy* is the release number. For example, z/OS V2R1.

JES-member-name

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

JES-type

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

CPC-name

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

CPC-serial

Serial number of the CPC.

proxy-name

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

server-name

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

total-items

Number of system definitions that were retrieved.

Example

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are included in CPC *CPC1*.

```
GET /zosmf/resttopology/systems/cpcName/CPC1 HTTP/1.1  
Host: zosmf1.yourco.com
```

Figure 305. Sample request to retrieve a list of systems included in a CPC

A sample response is shown in [Figure 306 on page 703](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "CPC1",
      "cpcSerial": "30104",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys10",
      "systemName": "sys10",
      "sysplexName": "plex2",
      "groupNames": "production",
      "url": "https://zosmf10.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY10",
      "jesType": "JES2",
      "cpcName": "CPC1",
      "cpcSerial": "30104",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "2"
}

```

Figure 306. Sample response from a request to retrieve a list of systems included in a CPC

Validate the status of the selected system

You can use this operation to check the connection status of a specified system which is managed through the z/OSMF Systems task.

HTTP method and URI path

```
GET /zosmf/services/systems/v1/validation/system
```

where:

- **system** identifies the nicknames of the selected systems to be validated.
- **validationType** identifies the types of statuses to validate for the selected system.

Query Parameters

Table 374. Validate Status Query Parameters		
Query Parameter	Required	Description
system	No.	To validate multiple systems, provide multiple system parameters, for example, system=SYSA&system=SYSB If no system is provided, then validate LocalSystemDefinition.
validationType	No.	To validate multiple statuses, provide multiple validationType parameters, for example, validationType=URLConnectivity&validationType=SSO If no validation Type is provided, all eligible statuses are validated. Eligible status types: <ul style="list-style-type: none"> • URLConnectivity (z/OSMF connectivity) • SSO (SSO enablement) • MVSSystem (Jesplex members status) • LocalSystemDefinition (Whether it exists in the system table)

Status types that are eligible for validation

URL Connectivity (connectivity for z/OSMF)

- Description: This is to check whether you can connect to the z/OSMF URL of the selected system from the local system.
- Prerequisites:
 - URL value exists. Otherwise: N/A.

SSO (Single sign-on)

- Description: This is to check whether you can single sign-on to the selected system from the local system.
- Prerequisites:
 - URL value exists. Otherwise: N/A.
 - SSO is enabled. Otherwise: N/A.

MVSSystem (Sysplex members status)

- Description: This is to check whether the status of the system in the local Jesplex is ACTIVE. If the status is ACTIVE, the return is "Successful", otherwise, the return is "Failed".

- Note: Currently this only supports validating the system in the local Jesplex.

LocalSystemDefinition (Whether existing in system table)

- Description: This is to check whether the local system definition exists in the system table and the z/OSMF URL is reachable.
- **Note:** This only supports validating the existence of the local system so that it can be validated with or without any system selected. Any system definition that has the same URL value as the current z/OSMF that is running is considered a local system definition.

Standard headers

None.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 410](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 412](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 410](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "LocalSystemDefinition": {
    "result": "validation-result",
    "validationType": "LocalSystemDefinition",
    "message": "validation-message"
  },
  "systems": {
    "system-nickname": [
      {
        "result": "validation-result",
        "validationType": "validation-type",
        "message": "validation-message"
      }
    ]
  }
}
```

Table 375. Validate System Json Responses		
Json Response	Required	Description
Object		

Table 375. Validate System Json Responses (continued)

Json Response	Required	Description
systems object[]	Yes	Array of systems with validation results.
nickname string	Yes	Nickname of the current system.
ValidationResults object[]	Yes	Array of status validation results.
validationType string	Yes	Current validation type: <ul style="list-style-type: none"> • URLConnectivity (connectivity of z/OSMF) • SSO (SSO enablement) • MVSSystem (Jespex members status) • LocalSystemDefinition (Whether it exists in the system table)
result string	Yes	Validation result of current status. <ul style="list-style-type: none"> • Successful • Failed • NA • conflict (only applies to MVSSystem status)
message string	No	Detailed message of the validation result.

Example

In the following example, the GET method is used to check systems selected. A sample request is shown in [Figure 307 on page 706](#).

```
GET /zosmf/services/systems/v1/validation/system?system=SYSA&system=SYSB&validationType=
URLConnectivity&validationType=SSO
```

Figure 307. Sample request to check systems selected

A sample response is shown in [Figure 308 on page 707](#).


```

{
  "systems": [
    {
      "nickname": "SYSA",
      "validationResults": [
        {
          "validationType": "SSO",
          "result": "Failed",
          "message": "IZUG459E: The SSO is enabled, but the system is not available for single sign-on
at this moment."
        },
        {
          "validationType": "URLConnectivity",
          "result": "n/a",
          "message": "IZUG460I: There is no z/OSMF URL configured for this system."
        },
        {
          "validationType": "MVSSystem",
          "result": "Failed",
          "message": "IZUG469E: MVS System is not active at this moment. Current MVS system status is:
DRAINED."
        }
      ]
    },
    {
      "nickname": "SYSB",
      "validationResults": [
        {
          "validationType": "SSO",
          "result": "Successful"
        },
        {
          "validationType": "URLConnectivity",
          "result": "Successful"
        }
      ]
    }
  ]
}

```

Figure 308. Sample response to check systems selected

Case Analysis

The following tables outline the validation types and possible results from a variety of different Sysplex systems. All examples are operating under the assumption that you are connecting to the z/OSMF1 UI or a REST API.

Table 376. SYS1 zOSMF Local System		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check the connection to the URL of the local system in the system table.
SSO connection	N/A	N/A
MVS system status	Successful	N/A
Local system definition	Successful/Failed	This validation only applies to the local system. If the return is "Failed", other plugins like z/OS console services and z/OSMF workflow services may fail.

Table 377. SYS2 zOSMF2		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check the connection to zOSMF2 from the local system.
SSO connection	Successful/Failed/NA	Check the SSO to zOSMF2 from the local system.
MVS system status	Successful/Failed/NA	Check the system status in JESPLex <ul style="list-style-type: none"> • Successful if ACTIVE • Failed if not ACTIVE • NA if not in the Local JESPLex

Table 378. SYS3		
Validation Type	Possible Result	Note
URL connection	N/A	There is no z/OSMF URL configured for this system.
SSO connection	N/A	There is no z/OSMF URL configured for this system.
MVS system status	Successful/Failed/NA	Check the system status in JESPLex <ul style="list-style-type: none"> • Successful if ACTIVE • Failed if not ACTIVE • NA if not in the Local JESPLex.

Table 379. SYS4 zOSMF3		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check connection to zOSMF3 from local system.
SSO connection	Successful/Failed/NA	Check SSO to zOSMF3 from local system.
MVS system status	NA	Not in the Local JESPLex.

Table 380. SYS5		
Validation Type	Possible Result	Note
URL connection	NA	There is no z/OSMF URL configured for this system.
SSO connection	NA	There is no z/OSMF URL configured for this system.
MVS system status	NA	Not in the Local JESPLex.

Check all MVS systems in the local plex

You can use this operation to obtain a list of the systems that are defined to a z/OSMF instance.

HTTP method and URI path

GET /zosmf/services/systems/v1/validation/plex

Query Parameter

Table 381. Query parameters to check all systems in the local plex		
Query Parameter	Required	Descriptiona
validationType	No	<p>To validate multiple statuses, provide multiple validationType parameters, for example, validationType=URLConnectivity&validationType=SSO</p> <p>If no validation Type is provided, all eligible statuses are validated.</p> <p>Eligible status types:</p> <ul style="list-style-type: none">• URLConnectivity (z/OSMF connectivity)• SSO (SSO enablement)• MVSSystem (Jesplex members status)• LocalSystemDefinition (Whether it exists in the system table)

Status types that are eligible for validation

URL Connectivity (connectivity for z/OSMF)

- Description: This is to check whether you can connect to the z/OSMF URL of the selected system from the local system.
- Prerequisites:
 - URL value exists. Otherwise: N/A.

SSO (Single sign-on)

- Description: This is to check whether you can single sign-on to the selected system from the local system.
- Prerequisites:
 - URL value exists. Otherwise: N/A.
 - SSO is enabled. Otherwise: N/A.

MVSSystem (Sysplex members status)

- Description: This is to check whether the status of the system in the local Jesplex is ACTIVE. If the status is ACTIVE, the return is "Successful", otherwise, the return is "Failed".
- Note: Currently this only supports validating the system in the local Jesplex.

LocalSystemDefinition (Whether existing in system table)

- Description: This is to check whether the local system definition exists in the system table and the z/OSMF URL is reachable.
- **Note:** This only supports validating the local system so that it can be validated with or without any system selected. Any system definition that has the same URL value as the current z/OSMF that is running is considered a local system definition.

Standard headers

None.

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 688.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 412.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 410.

If the request was successful, the response also includes the following JSON object:

```
{
  "LocalSystemDefinition": {
    "result": "validation-result",
    "validationType": "LocalSystemDefinition",
    "message": "validation-message"
  },
  "systems": {
    "system-nickname": [
      {
        "result": "validation-result",
        "validationType": "validation-type",
        "message": "validation-message"
      }
    ]
  }
}
```

Table 382. Validate System Json Responses

Json Response	Required	Description
Object		
systems object[]	Yes	Array of systems with validation results.
nickname string	Yes	Nickname of the current system.
ValidationResults object[]	Yes	Array of status validation results.
validationType string	Yes	Current validation type: <ul style="list-style-type: none"> • URLConnectivity (connectivity of z/OSMF) • SSO (SSO enablement) • MVSSystem (Jespex members status) • LocalSystemDefinition (Whether it exists in the system table)
result string	Yes	Validation result of current status. <ul style="list-style-type: none"> • Successful • Failed • NA • conflict (only applies to MVSSystem status)
message string	No	Detailed message of the validation result.

Example

In the following example, the GET method is used to Check the MVS System. A sample request is shown [Figure 309 on page 711](#).

```
GET /zosmf/services/systems/v1/validation/plex?validationType=MVSSystem
```

Figure 309. Sample request to Check all the MVS Systems in the local plex

A sample response is shown in [Figure 310 on page 712](#).

```

{
  "systems": [
    {
      "nickname": "SYSA",
      "validationResults": [
        {
          "validationType": "SSO",
          "result": "Failed",
          "message": "IZUG459E: The SSO is enabled, but the system is not available for single sign-on
at this moment."
        },
        {
          "validationType": "URLConnectivity",
          "result": "n/a",
          "message": "IZUG460I: There is no z/OSMF URL configured for this system."
        },
        {
          "validationType": "MVSSystem",
          "result": "Failed",
          "message": "IZUG469E: MVS System is not active at this moment. Current MVS system status is:
DRAINED."
        }
      ]
    },
    {
      "nickname": "SYSB",
      "validationResults": [
        {
          "validationType": "SSO",
          "result": "Successful"
        },
        {
          "validationType": "URLConnectivity",
          "result": "Successful"
        }
      ]
    }
  ]
}

```

Figure 310. Sample response to Check all the MVS Systems in the local plex

Case Analysis

The following tables outline the validation types and possible results from a variety of different Sysplex systems. All examples are operating under the assumption that you are connecting to the z/OSMF1 UI or a REST API.

Table 383. SYS1 zOSMF Local System		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check the connection to the URL of the local system in the system table.
SSO connection	N/A	N/A
MVS system status	Successful	N/A
Local system definition	Successful/Failed	This validation only applies to the local system. If the return is "Failed", other plug ins like z/OS console services and z/OSMF workflow services may fail.

<i>Table 384. SYS2 zOSMF2</i>		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check the connection to zOSMF2 from the local system.
SSO connection	Successful/Failed/NA	Check the SSO to zOSMF2 from the local system.
MVS system status	Successful/Failed/NA	Check the system status in JESPLex <ul style="list-style-type: none"> • Successful if ACTIVE • Failed if not ACTIVE • NA if not in the Local JESPLex

<i>Table 385. SYS3</i>		
Validation Type	Possible Result	Note
URL connection	N/A	There is no z/OSMF URL configured for this system.
SSO connection	N/A	There is no z/OSMF URL configured for this system.
MVS system status	Successful/Failed/NA	Check the system status in JESPLex <ul style="list-style-type: none"> • Successful if ACTIVE • Failed if not ACTIVE • NA if not in the Local JESPLex.

<i>Table 386. SYS4 zOSMF3</i>		
Validation Type	Possible Result	Note
URL connection	Successful/Failed	Check connection to zOSMF3 from local system.
SSO connection	Successful/Failed/NA	Check SSO to zOSMF3 from local system.
MVS system status	NA	Not in the Local JESPLex.

<i>Table 387. SYS5</i>		
Validation Type	Possible Result	Note
URL connection	NA	There is no z/OSMF URL configured for this system.
SSO connection	NA	There is no z/OSMF URL configured for this system.
MVS system status	NA	Not in the Local JESPLex.

TSO/E address space services

TSO/E address space services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with TSO/E address spaces on a z/OS system.

Table 388 on page 714 lists the operations that the TSO/E address space services provide.

Table 388. Operations provided through the TSO/E address space services.	
Operation	HTTP method and URI path
“Start or reconnect to a TSO/E address space” on page 718	POST /zosmf/tsoApp/tso?parms
“Start an application in a TSO/E address space” on page 722	POST /zosmf/tsoApp/app/servletKey/appKey
“Receive messages from a TSO/E address space” on page 742	GET /zosmf/tsoApp/tso/servletKey
“Receive messages from an application” on page 743	GET /zosmf/tsoApp/app/servletKey/appKey
“Send messages to a TSO/E address space” on page 737	PUT /zosmf/tsoApp/tso/servletKey?[readReply=true false]
“Send messages to an application” on page 739	PUT /zosmf/tsoApp/app/servletKey/appKey
“Ping a TSO/E address space” on page 740	PUT /zosmf/tsoApp/tso/ping/servletKey
“End a TSO/E address space” on page 744	DELETE /zosmf/tsoApp/tso/servletKey?[tsoforcecancel=true false]

How to use the Swagger interface

You can use the Swagger interface to display information about the TSO/E address space services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Required authorizations

Generally, your z/OSMF user ID requires the same authorizations for using the TSO/E address space services as when you perform these operations through a TSO/E session on the z/OS system. For

example, to start an application in a TSO/E address space, your user ID must be authorized to operate that application.

In addition, to use TSO/E address space services, you must have:

- READ access to the *account* resource in class ACCTNUM, where *account* is the value that is specified in the COMMON_TSO ACCT option in parmlib.
- READ access to the CEA.CEATSO.TSOREQUEST resource in class SERVAUTH.
- READ access to the *proc* resource in class TSOPROC, where *proc* is the value that is specified with the COMMON_TSO PROC option in parmlib.
- READ access to the <SAF_PREFIX>.*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF_PREFIX>.IzuManagementFacilityTsoServices.izuUsers resource name in the EJBROLE class.

You must also ensure that the z/OSMF started task user ID, which is IZUSVR by default, has READ access to the CEA.CEATSO.TSOREQUEST resource in class SERVAUTH.

To create a TSO/E address space on a remote system, you require the following authorizations:

- You must be authorized to the SAF resource profile that controls the ability to send data to the remote system (*systemname*), as indicated:

```
CEA.CEATSO.FLOW.systemname
```

- To flow data between different systems in the sysplex, you must be authorized to do so by your external security manager, such as a RACF database with sysplex-wide scope. For example, to flow data between System A and System B, you must have access to the following resource profiles:

```
CEA.CEATSO.FLOW.SYSTEMA  
CEA.CEATSO.FLOW.SYSTEMB
```

The TSO/E address space services authority might already be defined if you are using z/OS data set and file REST services, as those services require similar authority.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all TSO/E address space services for returning data and status about the requested operations. The attributes that are provided in the JSON object depend on the requested operation.

```
{  
  "servletKey": "servlet-key",  
  "ver": "structure-version",  
  "queueID": "message queue ID",  
  "remoteSys": "remote system",  
  "ceatsoconn": "CEA connection handler",  
  "tsoData": "TSO/E-messages",  
  "appData": "application-messages",  
  "timeout": "timeout-indicator",  
  "reused": "reconnected-indicator",  
  "msgData": "z/OSMF-messages",  
  "messages": "unexpected z/OSMF-messages"  
}
```

Where:

servletKey

Unique identifier for the servlet entry. It maps to the TSO/E address space ID and provides additional information about the address space. To communicate with the TSO/E address space, the client must provide the servlet key.

ver

Version of the TSO/E address space services and the JSON object structure that is used for this request. The version sequence starts at 0100, and is incremented only if the services or the JSON structure changes. In your application, check the value of the returned structure and verify that your application is compatible with the current API.

queueID

When the TSO/E address space interface starts a new TSO/E session, it also creates a new z/OS UNIX message queue to enable communication between the client and the TSO/E address space. This value is the identifier for the z/OS UNIX message queue.

remoteSys

System name of the remote system on which the TSO/E address space is to be started.

ceatsoconn

100-byte binary key (in hexadecimal), which is used by callers to perform subsequent operations with a remote TSO/E address space. To use this value with CEA APIs, the caller must first convert it to raw binary. This hexadecimal string includes 200 characters, where 2 characters represent 1 byte.

tsoData

TSO/E messages that were received during the request. The *tsoData* attribute is included in the JSON object only if TSO/E messages were received.

The value that is returned in the *tsoData* attribute is a JSON object that describes the messages that were received. For example, the TSO/E message JSON format has the following syntax:

```
{ "message-type": { "VERSION": "JSON-version", "data-type": "data-value" } }
```

where:

message-type

Keyword that identifies the type of TSO/E message. The value can be TSO MESSAGE, TSO PROMPT, or TSO RESPONSE.

JSON-version

A four-digit number that identifies the JSON version that is used to format the message.

data-type

Keyword that describes the type of data that is included in the data-value variable. The value can be DATA, HIDDEN, or ACTION.

Example: { "TSO RESPONSE": { "VERSION": "0100", "DATA": "ALLOC DA" } }

appData

Messages that are received from an application that is running in a TSO/E address space during the request. The *appData* attribute is included in the JSON object only if messages were received from an application and no TSO/E messages were received during the request.

timeout

Indicator of whether the request timed out while it waited for a response. The value is "true" if the request timed out. Otherwise, the value is "false".

If the service creates a new TSO/E address space, the service attempts to read the initial startup TSO/E messages. If no messages are received in the allotted time, this value is set to "true".

If the service reconnects the user to an existing TSO/E address space, no startup messages are expected; therefore, the service does not wait for any startup TSO/E messages.

reused

Indicator of whether the service connected the user to an existing TSO/E session instead of a new session. The *reused* attribute is included in the JSON object only if the *appsessid* parameter is provided for the start TSO/E address space request. The value that is returned for the *reused* attribute is "true" if a TSO/E address space with that *appsessid* exists. Otherwise, the value is "false".

msgData

z/OSMF messages received during the request. The messages attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

messages

z/OSMF messages received for unexpected errors.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the TSO/E address space services.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the TSO/E address space services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Usage Notes

- Establish the SYSCALL environment in your REXX script.

If you want to run a REXX script and get the response from the TSO/E REST APIs, establish the SYSCALL environment in your REXX script with a SYSCALL request that begins with ADDRESS SYSCALL. For example,

```
if syscalls('ON')>3 then
do
  say 'Unable to establish the SYSCALL environment'
  return
end
```

For more information on how to establish the SYSCALL environment, see [Establishing the SYSCALL environment](#) in *z/OS Using REXX and z/OS UNIX System Services*

- Things to consider when you use the TSO/E command ALLOCATE:

If you plan to use ALLOCATE to create a data set or file with the APIs provided, you need to use the TSO/E command FREE to release it after you finish working on it. Otherwise, the TSO/E address space locks the data set or file it creates. This is important if you are using the API PUT /zosmf/tsoApp/{version}/tsowith cmdState=stateless. Ensure that the ALLOCATE and FREE commands are invoked in pairs.

- How to use 8-character user ID:

By default, TSO/E address space services only support user IDs with 7 characters at the most. If you want to use TSO/E address space services with an 8-character user ID, update the USERIDMAX parameter for IKJTSOxx.

For more information, see [Statements and parameters for IKJTSOxx](#) in *z/OS MVS Initialization and Tuning Reference*.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Start or reconnect to a TSO/E address space

You can use this operation to start a new TSO/E address space or to reconnect to a dormant TSO/E address space.

HTTP method and URI path

```
POST /zosmf/tsoApp/tso?<parms>
```

Where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<parms>** qualifies the request with one or more of the parameters that are described in [Table 389](#) on page 718.

When the TSO/E address space interface starts a new TSO/E session, it also creates a new z/OS UNIX message queue to enable communication between the client and the TSO/E address space.

When the interface reconnects to a dormant TSO/E address space, the interface reuses the session resources, including the z/OS UNIX message queue.

Note: A *dormant TSO/E address space* is an address space that is deactivated for communication through its z/OS UNIX message queue, but remains available at a TSO/E READY prompt for time.

Supported parameters

Table 389. Supported parameters for the start and reconnect TSO/E session requests		
Parameter	Required	Description
proc	Yes	Name of the TSO/E logon procedure to use to log in to the TSO/E address space.
chset	Yes	Character set to use for the caller's TSO/E address space. This value is used by the applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC. The default character set, which is 697 decimal, are used if zero is specified as the value.
cpage	Yes	Code page to use for the caller's TSO/E address space. This value is used by the applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC. The default code page, which is 1047 decimal, is used if zero is specified as the value.
rows	Yes	Number of rows to be displayed on the screen. The default number of rows, which is 24, is used if zero is specified as the value.

Table 389. Supported parameters for the start and reconnect TSO/E session requests (continued)

Parameter	Required	Description
cols	Yes	Number of columns to be displayed on the screen. The default number of columns, which is 80, is used if zero is specified as the value.
acct	No	TSO/E user account number.
ugrp	No	Name of the TSO/E user group.
rsize	No	Region size to use for the TSO/E address space.
appsessid	No (for new), Yes (for reconnect)	Identifier that uniquely identifies the TSO/E address space. This parameter is optional when starting a new TSO/E address space, and it is required when reconnecting to an existing TSO/E address space. If an address space with the specified identifier does not exist, a new TSO/E address space is created and assigned the identifier that is specified.
system	No	System on which the TSO/E address space is to be created or reconnected. Specify the system name. This parameter is optional; if not specified, the request is processed on the local system.
apptag	No	Identifies the application that is responsible for creating the TSO/E address space. This value is used by CEA. The value of apptag is 2 - 8 characters (A-Z, a-z, 0-9) and cannot begin with a digit. The value is case-sensitive. This parameter is optional; if not specified, the default value is IZUTSOAP.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 714.

In addition, only the z/OSMF user that started the TSO/E address space is authorized to use the z/OS UNIX message queue that is associated with that address space.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object with additional information about the results of the request. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example: Creating a new TSO/E address space on the local system

To create a new address space with the following settings on the local system, submit the request that is depicted in [Figure 311 on page 720](#):

- Procedure name: IKJACCNT
- Character set: 697
- Code page: 1047
- Screen rows: 204
- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT

```
POST /zosmf/tsoApp/tso?proc=IKJACCNT&chset=697&cpage=1047&rows=204
&cols=160&rsize=50000&acct=DEFAULT HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 311. Sample request to create a new TSO/E address space on the local system

A sample response is shown in [Figure 312 on page 720](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-SYS1-55-aaakaaac","queueID":"4","sessionID":"0x37",
"ver":"0100","tsoData":[{"TSO MESSAGE":{"VERSION":"0100","DATA":"ZOSMFAD LOGON IN
PROGRESS AT 01:12:04 ON JULY 17, 2017"}}], "reused":false,"timeout":false}
```

Figure 312. Sample response from create TSO/E address space request on the local system

Example: Creating a new TSO/E address space on a remote system

To create a new address space with the following settings on a remote system (SYS2), submit the request that is depicted in [Figure 313 on page 721](#):

- Procedure name: IKJACCNT
- Character set: 697
- Code page: 1047
- Screen rows: 204
- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT
- System: SYS2

Host: zosmf1.yourco.com

A sample response is shown in Figure 314 on page 721.

[illegible]

Example: Reconnecting to an existing TSO/E address space on the local system

- Procedure name: IKJACCNT
- Character set: 697
- Code page: 1047
- Screen rows: 204
- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT
- Application Session ID: sdsf 23715376543765

Host: zosmf1.yourco.com

A sample response is shown in Figure 316 on page 721.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-55-aaakaaac","queueID":"4","sessionId":"0x37","ver":"0100" "reused":false,"timeout":false}
```

Chapter 1. Using the z/OSMF REST services **721**

Content-Type: application/json

Custom headers

None.

Request content

To start an application in a TSO/E address space, your request must include a JSON object that contains the application start command, plus any input values that the application requires on start-up. The following example shows a possible JSON object structure. Note, however, that command requirements can vary. For the specific requirements for the start command, refer to the documentation for the command.

```
{"startcmd": "{command} &1 &2 {queueid} {user-parm-list}"}
```

Figure 319. Starting an application: example of the request content

In the example, the first three parameters are required to establish communication between the REST-caller and the command. The rest is optional. This order is not enforced.

The parameters are further described, as follows:

{command}

TSO/E command that is used to start the application. For example, the ISFWEB parameter is used to start the SDSF application.

&1 and &2

Variables for passing the message type IDs to the TSO application that are represented by the “command”. TSO/E address space services use z/OS UNIX message queue to transfer information between REST-call and the TSO application. Message type IDs are needed when the TSO application reads messages from or write messages to the z/OS UNIX message queue. Before this REST API actually starts the “command”, &1 and &2 will be replaced by the actual message type IDs, respectively. The TSO application uses the value of &1 as the message type ID to write messages into the message queue. The value of &2 is the message type ID that the TSO application reads messages from the message queue. Currently, the value of &1 and &2 are hard coded as 4 and 32772 respectively.

{queueid}

Identifier for the z/OS UNIX message queue, which is used to enable communication between the client and the TSO/E address space. If you use the TSO/E address space services to start a new TSO/E session, the message queue ID is returned in the HTTP response body, along with other values. See [“Content type used for HTTP request and response data” on page 715](#).

{user-parm-list}

Optional list of application-specific parameters.

Processing overview

When the client requests to start an application in a TSO/E address space, the API completes the following actions:

- Assigns the command processor input and output message types to use for communication with the application to be started. The message types will be used only for the application identified by the **appKey**.
- Replaces the variables in the command with the assigned command processor message types.
- Sends the TSO/E command to the TSO/E address space identified by the servlet key.

- Attempts to read a TSO/E or application response message. If no messages are received in the time allotted, a timeout indication is returned. Any TSO/E messages are prioritized over application messages. Typically, when a caller receives a TSO/E message while attempting to receive application messages, the caller processes the TSO/E messages, then attempts to retrieve the queued application messages.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations”](#) on page 714.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 717.

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 715.

Example

The application *BkApp001* is a TSO/E command processor. To start this application in the TSO/E address space associated with servlet key *ZOSMFAD-71-aabcaaaaf*, submit the request shown in [Figure 320](#) on page 724.

```
POST /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com

{"startcmd": "ISFWEB &1 &2 12345"}
```

Figure 320. Sample request to start an application in a TSO/E address space

In this example, the request content includes three variables:

- Command processor output message type ID is passed in placeholder variable *&1*
- Command processor input message type ID is passed in placeholder variable *&2*
- Message queue ID is included as the third parameter *12345*.

Thus, if the start command is **ISFWEB** and the message queue ID was 12345, your request would specify the following values to start the application:

```
ISFWEB &1 &2 12345
```

Before the variables are passed to the command, *&1* is automatically resolved by z/OSMF to the message type that the application uses to send messages to the client (the output message) and *&2* is automatically resolved to the message type that is used by the application to receive messages from the client (the input message).

A sample response is shown in [Figure 321](#) on page 725.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100",
"appData":...,"timeout":false}
```

Figure 321. Sample response from a start an application in a TSO/E address space request

Issue a TSO/E command with z/OSMF REST API

You can use this operation to issue a TSO/E command and get a corresponding response.

HTTP method and URI path

```
PUT /zosmf/tsoApp/{version}/tso
```

Where:

- `/zosmf/tsoAPP` identifies the TSO/E address space services.
- `version` identifies the version of the TSO/E REST API service. The following value is valid: `v1`.
- `/tso` informs the service that the request is for a TSO/E address space.

Query parameters

None.

Standard headers

Use the following standard HTTP header with this request: `Content-Type: application/json`.

Request body

Table 390. Supported parameters for the TSO/E command with z/OSMF REST API.

The request content is expected to contain a JSON object. See [Table 390 on page 725](#) for a description of the fields.

Parameter	Required or Optional	Description
tsoCmd	Required	Specifies the command to issue.
system	Optional	Name of the system in the same sysplex that the command is routed to. The default is the local system.

Table 390. Supported parameters for the TSO/E command with z/OSMF REST API.

The request content is expected to contain a JSON object. See [Table 390 on page 725](#) for a description of the fields.

(continued)

Parameter	Required or Optional	Description
maxWaitTime	Optional	<p>Specify the max amount of time in seconds that the Issue command API will continue to get responses. The API tries to get as many responses as possible until the amount of time elapsed or until the TSO PROMPT is received. The TSO PROMPT indicates that the TSO address space is ready for more input from the client. The input can either be a new command or the next part of the previous command. After the TSO PROMPT is received, there is no further response for a TSO command.</p> <ul style="list-style-type: none"> • If you do not specify the value for maxWaitTime, the API gets as many messages as possible until any of the three things happen: <ul style="list-style-type: none"> – The TSO PROMPT is received. – The API returns if there is no new response during last 5 seconds. – 30 seconds elapses. • The minimal value is 1 second. • The maximum value is 300, which means 5 mins.

Table 390. Supported parameters for the TSO/E command with z/OSMF REST API.

The request content is expected to contain a JSON object. See [Table 390 on page 725](#) for a description of the fields.

(continued)

Parameter	Required or Optional	Description
cmdState	Optional	<p>Specify the state of the TSO/E command you want to issue:</p> <p>stateless</p> <p>The tsoCmd is a stateless TSO/E command. You can use a single 'Issue TSO/E command REST API' to fulfill the request.</p> <p>Note: for a stateless command, all the command responses are returned in the API response. You cannot perform any further action against the TSO/E address space, which is used to issue the command. This is the default. For a single user, you can issue up to 45 stateless command APIs concurrently. Case is not significant.</p> <p>stateful</p> <p>The tsoCmd is a stateful TSO/E command, which means you need to issue a series of TSO/E commands that are related to each other. You need to issue them one after another by the Issue command REST API. Case is not significant.</p> <p>The address space serves these stateful commands, which are identified by the servletKey and kept alive for 10 minutes. If there is no further Issue stateful command or get response request from the user during that time, the address space is released.</p> <p>For a single user, you can issue up to 45 stateful command APIs concurrently. The API is differentiated by the servletKey. It is active for 10 minutes, and is counted as 1 of the 45 during that 10 minutes. This is true even if you do not issue any stateful commands/get response API in the TSO/E address space that is identified by the servletKey.</p>
servletKey	Optional	<p>Unique identifier for a stateful TSO/E command entry. It maps to the TSO/E address space in which the stateful TSO/E command is issued. You can get servletKey from the response of the Issue TSO/E command API. The servletKey is only valid if you previously issued a stateful TSO/E command with Issue Command REST API. For the first command of a serial of stateful TSO/E commands, you do not specify servletKey. You get the value of servletKey from the response JSON. The servletKey is invalid if you do not specify cmdState as stateful.</p>

Table 390. Supported parameters for the TSO/E command with z/OSMF REST API.

The request content is expected to contain a JSON object. See [Table 390 on page 725](#) for a description of the fields.

(continued)

Parameter	Required or Optional	Description
keyword	Optional	Specifies a regular expression that you want to detect in the command response. For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key: a[\s]+regular

Note:

- If you use a stateful Issue command API, you can use the servletKey to lock a TSO/E address space for following stateful commands/get response for 600 seconds. The address space, which is identified by servletKey will be released after 600 seconds if there's no further Issue stateful command/get response request from user.
- z/OSMF TSO/E services use CEA to create and maintain TSO/E address spaces. The default value of MAXSESSPERUSER for CEA is 10, which means CEA can create and maintain up to 10 TSO/E address spaces for a single user. The default value of MAXSESSIONS is 50, which means CEA can create and maintain up to 50 TSO/E address spaces on the system. You can issue command **f cea,d,parms** to check these values on your system. The maximum value for MAXSESSPERUSER is 99, and for MAXSESSIONS it is 2000. z/OSMF TSO/E services can maintain up to 45 TSO/E address spaces for stateless request, and up to 45 TSO/E address spaces for stateful request per user. You need to change the MAXSESSPERUSER to 99 and update the MAXSESSIONS to achieve 45/45 address spaces.
- For more information about how to change a CEA parameter, see [Working with TSO/E address spaces started by CEA in z/OS MVS Programming: Callable Services for High-Level Languages and CEAPRMxx \(common event adapter parameters\) in z/OS MVS Initialization and Tuning Reference](#).

Required authorizations

To issue the TSO/E command and get response by the TSO/E address space services, you must have:

- – READ access to resource account in class ACCTNUM, where account is the value that is specified in the COMMON_TSO ACCT option in parmlib.
- READ access to resource proc in class TSOPROC, where proc is the value that is specified with the COMMON_TSO PROC option in parmlib.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request is complete. Status code 200 indicates success. A status code of 4nn or 5nn indicates an error. For more information, see [“Error handling” on page 717](#).

Table 391. Response content for a successful issue command request

Field name	Description
cmdResponse	Command response in a JSON array.

Table 391. Response content for a successful issue command request (continued)	
Field name	Description
servletKey	Unique identifier for the servlet entry. It maps to the TSO/E address space in which the TSO/E command is issued. servletKey is returned only when cmdState is stateful.
tsoPromptReceived	Whether the TSO PROMPT sign is received in the command response: Y TSO PROMPT is received. N TSO PROMPT is not received yet.
keywordDetected	The result of the response detection request. This is returned when the keyword is specified. The values are: Y Matching record in the response was found. N Matching record in the response was not found.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 392. Response JSON object for an unsuccessful issue command request	
Field name	Description
returnCode	Identifies the category of error.
returnCode	Identifies the specific error.
reason	Text that describes the cause of the error.

HTTP status codes

For a successful request, HTTP status code 200 is returned, and the response body is provided, as described in [Table 391 on page 728](#).

For unsuccessful requests, the service returns the status codes that are described in [Table 392 on page 729](#).

Table 393. Response codes for unsuccessful issue command requests				
HTT P Stat us	Return code	Reaso n code	Reason	Description
500	8	4	An error occurred in the TSO/E address space. The error description: %s	The request failed because an error occurred. The error description is provided in the message text: %s. To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.

Table 393. Response codes for unsuccessful issue command requests (continued)

HTT P Stat us	Return code	Reason code	Reason	Description
500	8	12	The system cannot get the local node name and cannot start the TSO/E address space.	The attempt to prepare a TSO/E address space failed. The z/OSMF TSO/E service failed to retrieve the local node name of the system. Retry the request. If the problem persists, contact your z/OSMF administrator.
400	8	13	Unsupported Encoding Exception: %s	The system cannot support %s encoding. For more information, check the z/OSMF logs. Retry the request. If the problem persists, contact the IBM Support Center and provide the error details.
400	8	14	An error occurred when parsing the TSO/E response data.	The z/OSMF TSO/E service failed to parse the TSO/E response data. For more information, check the z/OSMF logs. Retry the request. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	16	An I/O Exception occurred when parsing the issue command request JSON body.	The z/OSMF TSO/E service failed to parse the issue command request JSON body. For more information, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	17	The TSO/E API cannot recognize the json field: %s	The JSON field %s in the request body, is not a supported field.
500	8	18	The tsoCmd cannot be null.	The JSON field “tsoCmd” in the request body cannot be null.
500	8	19	The maxWaitTime must be a positive integer between 1 to 300, cannot be %d.	The value of the “maxWaitTime” in the request JSON body must be a positive integer 1 - 300 .
500	8	21	The servletKey cannot be null.	The JSON field “servletKey” in the request body cannot be null.

Table 393. Response codes for unsuccessful issue command requests (continued)

HTTP Status	Return code	Reason code	Reason	Description
500	8	22	The maximum number of TSO/E address spaces that are allowed for the current user has been reached.	<p>Refer to message IZUG1127E for a detailed explanation.</p> <p>Display the active TSO/E address spaces and remove or cancel any address spaces that the user no longer needs.</p> <p>To display the active TSO/E address spaces, enter the command D TS,ALL from the operator console.</p> <p>To cancel a TSO/E address space, issue the C u=user-ID,a=ASID command from the operator console, where user-ID is the user's TSO/E ID and ASID is the address space identifier.</p>
500	8	23	No TSO/E address space exist for servletKey:%s. Check your servletKey.	No TSO/E address space exists for servletKey:%s. Check your servletKey in the request JSON body.
500	8	25	Your attempt to start a TSO/E address space for your command has timed out. Try again later.	Your attempt to start a TSO/E address space for your command timed out. Try again later.
500	8	26	The TSO/E address space is temporarily unavailable. Refer to IZUG1113E for details and try again later.	The TSO/E address space is temporarily unavailable. Refer to IZUG1113E for details and try again later.
500	8	27	The TSO/E address space cannot be created, %s.	The TSO/E address space cannot be created. The context of the error is provided in the message text: %s. Refer to IZUG1117E for details.
500	8	28	System I/O exception.	System I/O exception. To obtain more details about the error, check the z/OSMF logs.
500	8	29	The TSO/E address space could not be created because an error occurred with the logon procedure or the user settings.	Verify that the logon procedure exists and is valid. For more information, see message IZUG1121E.
500	8	30	The TSO/E address space for the request cannot be found.	The TSO/E address space for the request cannot be found. For more information, check the z/OSMF logs.

Table 393. Response codes for unsuccessful issue command requests (continued)

HTT P Stat us	Return code	Reason code	Reason	Description
500	8	31	The maximum number of TSO/E address spaces for the system has been reached.	<p>Refer to IZUG1105E for a detailed explanation.</p> <p>Display the active TSO/E address spaces and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command D TS, ALL from the operator console.</p> <p>To cancel a TSO/E address space, enter the command C u=user-ID,a=ASID from the operator console, where user-ID is the user's TSO/E ID and ASID is the address space identifier.</p>
500	8	32	Failed to create a TSO/E address space. TSO/E user account number has not been defined for use, %s	<p>Refer to message IKJ56486I for a detailed explanation.</p> <p>One of the following scenarios occurred:</p> <ul style="list-style-type: none"> • The specified account number is not defined to the RACF database. <ul style="list-style-type: none"> – The RACF administrator must first define the account number as a RACF resource and then give the user access that uses the PERMIT command. However, if the procedure is not in the procedure library, the logon attempt continues to fail. • The RACF class ACCTNUM is not active. <ul style="list-style-type: none"> – The RACF administrator must activate the RACF class.
500	8	33	Failed to create a TSO/E address space. TSO/E user account number has not been authorized for the user, %s	<p>Refer to message IKJ56487I for a detailed explanation.</p> <p>The specified account number is defined to the RACF database. However, this particular user ID is not allowed to use it.</p>
500	8	34	Failed to create a TSO/E address space. TSO/E user account number is invalid: %s	<p>Refer to message IKJ56702I for a detailed explanation. The specified account number is incorrect.</p>

Table 393. Response codes for unsuccessful issue command requests (continued)

HTTP Status	Return code	Reason code	Reason	Description
500	8	35	Failed to create a TSO/E address space. Following messages returned by system: %s	z/OSMF TSO/E service failed to create TSO/E address space. The context of the error is provided in the message text: %s. To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	36	Failed to create a TSO/E address space. %s	z/OSMF TSO/E service failed to create TSO/E address space due to the region size exceeds the limit size. The context of the error is provided in the message text: %s. To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	37	The system you specified is incorrect.	Specify your request with the correct name of the target system and try again.
500	8	38	The maximum number of TSO/E address spaces that are allowed for the current user to issue a stateless command has been reached.	For a single user, you can have up to 45 active TSO/E address space to issue a stateless command.
500	8	39	The maximum number of TSO/E address spaces that are allowed for the current user to issue a stateful command has been reached.	For a single user, you can have up to 45 active TSO/E address space to issue stateful command.

Example

In the following example, the PUT method is used to issue a TSO/E command **TIME** and get corresponding response. On completion, the command response is returned to the user.

```
Request:
PUT https:// your.company.com/zosmf/tsoApp/v1/tso

Request body:
{
  "tsoCmd" : "TIME"
}

Response:
HTTP/1.1 200 OK
{
  "cmdResponse": [
    {
      "message": "TIME-07:35:12 AM. CPU-00:00:00 SERVICE-21 SESSION-00:00:01 APRIL 6,2021"
    },
    {
```

```

    {
      "message": "READY "
    }
  ],
  "tsoPromptReceived": "Y"
}

```

Get the response to a command that was issued with the TSO/E REST API

This API gets a response for the stateful TSO/E command. This means you set cmdState to stateful for a previous Issue command REST API.

HTTP method and URI path

```
GET /zosmf/tsoApp/{version}/tso
```

Where:

- zosmf/tsoApp identifies the TSO/E address space services.
- {version} identifies the version of the TSO/E REST API service. The following value is valid: v1.
- tso informs the service that the request is for a TSO/E address space.

Query parameters

Table 394.		
Parameter	Required or Optional	Description
servletKey	Required	A unique identifier for the servlet entry. It maps to the TSO/E address space in which the TSO/E stateful command is issued. You can get a servletKey from the response of the Issue TSO/E command API.
maxWaitTime	Optional	<p>Specifies the max amount of time in seconds that the Get response API continue to get responses. The API tries to get as many responses as possible until a certain amount of time elapses or the TSO PROMPT is received. TSO PROMPT indicates that the TSO/E address space is ready for more input from the client. The input can either be a new command or the next part of the previous command. There's no further response for a TSO/E command after the TSO PROMPT is received.</p> <ul style="list-style-type: none"> • If you do not specify the value for maxWaitTime, the API gets as many messages as possible until any of the following three things happen: <ul style="list-style-type: none"> – The TSO PROMPT is received. – The API returns no new response during the last 5 seconds. – 30 seconds elapses. • The minimal value is 1 second. • The maximum value is 300, which means 5 mins.

<i>Table 394. (continued)</i>		
Parameter	Required or Optional	Description
keyword	Optional	Specifies a regular expression that you want to detect in the command response. For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key: a[\s]+regular
sourceCmd	Optional	Specifies whether the API returns the TSO/E command, which was issued previously by Issue command request. Case is not significant. Y Return tsoCmd in the response JSON. N Do not return tsoCmd in the response JSON. This is the default.

Required authorizations

See the Required authorizations section of [“Issue a TSO/E command with z/OSMF REST API”](#) on page 725.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request is complete. Status code 200 indicates success. A status code of 4nn or 5nn indicates an error.

<i>Table 395. Response content for a successful get response request</i>	
Field	Description
cmdResponse	Command response in a JSON array.
tsoPromptReceived	Whether the TSO PROMPT sign is received in the command response: Y TSO PROMPT is received. N TSO PROMPT is not received yet.
keywordDetected	Result of the response detection request. Returned when keyword is specified. The values are: Y Matching record in the response was found. N Matching record in the response was not found.
sourceCmd	The TSO/E command, which was issued by the Issue TSO/E command request previously. Returned when sourceCmd is Y.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 396. Response JSON object for an unsuccessful issue command request	
Field	Description
returnCode	Identifies the category of the error.
reasonCode	Identifies the specific error.
reason	Text that describes the cause of the error.

HTTP status codes

For a successful request, the HTTP status code 200 is returned and the response body is provided, as is described in [Table 395 on page 735](#).

For unsuccessful requests, see the HTTP status codes section within [“Issue a TSO/E command with z/OSMF REST API” on page 725](#).

Example

In this example, you start a REXX script with the stateful PUT `/zosmf/tsoApp/{version}/tso API` and get part of the script output first. You then get the rest of the output with GET `/zosmf/tsoApp/{version}/tso API`.

- REXX script example: ZOSMF.JINGHUA.REXX(RXSAY)

```
- /* REXX */
/******
/*
/* Invoke like
/* exec 'ZOSMF.JINGHUA.REXX(RXSAY)' '10 1'
/*
/******
parse arg loopCnt sleepTime
do i = 1 to loopCnt
  say i "of" loopCnt
  CALL SYSCALLS('ON') /*ENABLE USS-CALLS*/
  ADDRESS SYSCALL
  "SLEEP" sleepTime /*SLEEP FOR ?? SECONDS*/
  CALL SYSCALLS 'OFF'
end
```

- Issue command

```
- Request:
PUT https:// your.company.com/zosmf/tsoApp/v1/tso

Request body:
{
  "tsoCmd" : "exec 'ZOSMF.JINGHUA.REXX(RXSAY)' '10 1'",
  "maxWaitTime" : "1",
  "cmdState" : "stateful"
}
Response:
HTTP/1.1 200 OK
{
  "cmdResponse": [
    {
      "message": "1 of 10"
    },
    {
      "message": "2 of 10"
    }
  ],
  "servletKey": "ZOSMFT1-31-aaauaaad",
  "tsoPromptReceived": "N"
}
```

- Get response.

```

- GET /zosmf/tsoApp/v1/tso?servletKey=ZOSMFT1-31-aaauaad&maxWaitTime=15
Response body:
{
  "cmdResponse": [
    {
      "message": "3 of 10"
    },
    {
      "message": "4 of 10"
    },
    {
      "message": "5 of 10"
    },
    {
      "message": "6 of 10"
    },
    {
      "message": "7 of 10"
    },
    {
      "message": "8 of 10"
    },
    {
      "message": "9 of 10"
    },
    {
      "message": "10 of 10"
    },
    {
      "message": "READY "
    }
  ],
  "tsoPromptReceived": "Y"
}

```

Send messages to a TSO/E address space

You can use this operation to send TSO/E messages to a TSO/E address space.

HTTP method and URI path

```
PUT /zosmf/tsoApp/tso/<servletKey>?[readReply=true|false]
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space to which the message will be sent.
- **[readReply]** is an optional parameter that indicates whether the service should send the message and immediately check for a response (default) or just send the message. To immediately check for a response, omit the parameter or set its value to *true*. Otherwise, set its value to *false*.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

Your request must include a JSON object that describes the message to be sent. For example, the TSO/E message JSON format has the following syntax:

```
{ "message-type": { "VERSION": "JSON-version", "data-type": "data-value" } }
```

where:

message-type

Keyword that identifies the type of TSO/E message. The value can be TSO MESSAGE, TSO PROMPT, or TSO RESPONSE.

JSON-version

A four-digit number that identifies the JSON version used to format the message.

data-type

Keyword that describes the type of data included in the *data-value* variable. The value can be DATA, HIDDEN, or ACTION.

Example: { "TSO RESPONSE": { "VERSION": "0100", "DATA": "ALLOC DA" } }

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

In addition, note that the API will attempt to read response TSO/E messages after the input message is sent. If no TSO/E messages are received after a predetermined time period, a timeout indication will be returned.

Required authorizations

See [“Required authorizations”](#) on page 714.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 717.

The response also includes a JSON object that contains the TSO/E response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 715.

Example

To send a TSO/E message to the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaaaf* and read the response TSO/E messages, submit the request depicted in [Figure 322](#) on page 738.

```
PUT /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com

{ "TSO RESPONSE": { "VERSION": "0100", "DATA": "TIME" } }
```

Figure 322. Sample request to send a message to a TSO/E address space

A sample response is shown in [Figure 323](#) on page 739.


```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100","tsoData":[{"TSO MESSAGE":
{"VERSION":"0100","DATA":"TIME-12:09:07 PM. CPU-00:00:00 SERVICE-92319
SESSION-00:00:13 OCTOBER 12,2011"}]}],"timeout":false}
```

Figure 323. Sample response from send message to TSO/E address space request

Send messages to an application

You can use this operation to send messages to an application running in a TSO/E address space.

HTTP method and URI path

```
PUT /zosmf/tsoApp/app/<servletKey>/<appKey>
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **app** informs the service that the request is for an application running in a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space where the application is running.
- **<appKey>** identifies the application to which to send messages.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: text/plain

Custom headers

None.

Request content

Your request must include a JSON object that contains the application message to be sent.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, note that the API will attempt to read application and TSO/E response messages after the application input message is sent. If no messages are received in the time allotted, a timeout indication will be returned.

Any TSO/E messages are prioritized over application messages. Typically, when a caller receives a TSO/E message while attempting to receive application messages, the caller processes the TSO/E messages, then attempts to retrieve the queued application messages.

Required authorizations

See [“Required authorizations” on page 714](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

To send a message to application *BkApp001*, which is running in the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 324 on page 740](#).

```
PUT /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com

' ... '
```

Figure 324. Sample request to send a message to an application

A sample response is shown in [Figure 325 on page 740](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","appData":[...],"timeout":false}
```

Figure 325. Sample response from send message to an application request

Ping a TSO/E address space

You can use this operation to ping a TSO/E address space. Doing so at regular intervals helps to ensure that the TSO/E address space remains active for the client. Otherwise, the server can end the TSO/E address space without warning.

HTTP method and URI path

```
PUT /zosmf/tsoApp/tso/ping/<servletKey>
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **ping** informs the service to ping the specified TSO/E address space.
- **<servletKey>** identifies the TSO/E address space for the service to ping.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Each TSO/E address space has an idle application time that the TSO/E address space services interface uses to determine whether the client application that is associated with the address space is active. If the idle application time is 10 minutes, the client application is considered to be inactive. In which case, the API ends all the TSO/E address spaces associated with the client application.

To prevent TSO/E address spaces from ending because of idle application time, callers can issue a ping request at least once every 5 minutes. Doing so informs the TSO/E address space services interface that the client application is still active, and causes the interface to reset the idle application time for all the TSO/E address spaces associated with the client application.

Required authorizations

See [“Required authorizations”](#) on page 714.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 717.

The response also includes a JSON object that contains all the attributes in the JSON structure except the message data. For more details, see [“Content type used for HTTP request and response data”](#) on page 715.

Example

To ping the TSO/E address space identified by servlet key `ZOSMFAD-71-aabcaaaaf`, submit the request depicted in Figure 326 on page 741.

```
PUT /zosmf/tsoApp/tso/ping/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 326. Sample request to ping a TSO/E address space

A sample response is shown in Figure 327 on page 741.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100","timeout":false}
```

Figure 327. Sample response from ping TSO/E address space request

Receive messages from a TSO/E address space

You can use this operation to receive messages from a TSO/E address space.

HTTP method and URI path

```
GET /zosmf/tsoApp/tso/<servletKey>
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space from which to receive messages.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

In addition, note that the API will attempt to read TSO/E messages. If no TSO/E messages are received after 15 seconds, a timeout indication will be returned.

Required authorizations

See [“Required authorizations”](#) on page 714.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 717.

The response also includes a JSON object that contains the TSO/E response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 715.

Example

To read TSO/E messages from the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaaaf*, submit the request depicted in [Figure 328](#) on page 742.

```
GET /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 328. Sample request to receive a message from a TSO/E address space

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","tsoData":
[{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> LOGON proc version = 04/28/2011"}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"  "}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> System Name = DCEIMGNE"}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> System Suffix = NE"}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> SYSPLEX Name = CFCIMGNE"}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> SYSRES Volume = SD1131"}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"  "}}],
"timeout":false}
```

You can use this operation to receive messages from an application running in a TSO/E address space.

```
GET /zosmf/tsoApp/app/<servletKey>/<appKey>
```

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **app** informs the service that the request is for an application running in a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space where the application is running.
- **<appKey>** identifies the application to which to send messages.

Use the following standard HTTP header with this request:

Custom headers

Request content

Usage considerations

In addition, note that the API will attempt to read application and TSO/E response messages. If no messages are received in the time allotted, a timeout indication will be returned.

Chapter 1. Using the z/OSMF REST services **743**

Required authorizations

See [“Required authorizations” on page 714](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

To receive TSO/E or application messages from application *BkApp001*, which is running in the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 330 on page 744](#).

```
GET /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 330. Sample request to receive messages from an application

A sample response is shown in [Figure 331 on page 744](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","appData":[...],"timeout":false}
```

Figure 331. Sample response for request to receive messages from an application

End a TSO/E address space

You can use this operation to end a TSO/E address space or place it in a dormant state as a candidate for reconnection.

HTTP method and URI path

```
DELETE /zosmf/tsoApp/tso/<servletKey>?[tsoforcecancel=true|false]
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space to be ended or placed in a dormant state.
- **[tsoforcecancel]** is an optional parameter that indicates whether to use the CANCEL or LOGOFF command to end the TSO/E address space. The parameter can have one of the following values:
 - **True:** The CANCEL command will be issued and the TSO/E address space will not be placed in a dormant state.
 - **False** (default): The LOGOFF command will be issued. If the CEA reconnect feature is enabled in your installation, the TSO/E address space will be placed in a dormant state. Otherwise, the TSO/E session will end.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 714](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object with additional information about the results of the request. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example: Logging off a TSO/E address space

To use the LOGOFF command to end the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 332 on page 745](#).

```
DELETE /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaf HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 332. Sample request to logoff a TSO/E address space

A sample response is shown in [Figure 333 on page 745](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","timeout":false,"reuse":false}
```

Figure 333. Sample response for logoff a TSO/E address space request

Example: Canceling a TSO/E address space

To use the CANCEL command to end the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 334 on page 746](#).

```
DELETE /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaf?tsoforcecancel=true HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 334. Sample request to cancel a TSO/E address space

A sample response is shown in Figure 335 on page 746.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","timeout":false,"reuse":true}
```

Figure 335. Sample response for a cancel TSO/E address space request

WLM resource pooling services

The WLM resource pooling services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. The WLM resource pooling services provide a programming interface for WLM policy elements. You can work with WLM policy elements in the context of IBM Cloud Provisioning and Management for z/OS.

With the WLM resource pooling services, you can provision and deprovision WLM policy elements, dynamically construct a new service definition, and install the service definition.

Table 397 on page 746 lists the operations that the WLM resource pooling services provide.

Table 397. Operations provided through the WLM resource pooling services.	
Operation	HTTP method and URI path
“Prime a WLM resource pool” on page 747	POST /zosmf/zwlm/rest/wrps
“Delete a WLM resource pool” on page 749	DELETE /zosmf/zwlm/rest/wrps/wrpid
“Construct a WLM service definition” on page 751	PUT /zosmf/zwlm/rest/policy/inspolicy
“Construct a WLM service definition with remove and install” on page 753	PUT /zosmf/zwlm/rest/policy/inspolicy

Required authorizations

The user’s z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.RESOURCE_POOL.WLM.*domainid*, where *domainid* is the identifier of the domain of systems.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. The HTTP status codes are described in the topics for the individual services.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 204 No Content

Request was processed successfully.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

Error logging

Errors from the WLM resource pooling services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Prime a WLM resource pool

Use this operation to create a record for a WLM resource pool.

HTTP method and URI path

```
POST /zosmf/zwlm/rest/wips
```

Query parameters

None.

Description

This operation creates a WRP record in a persistent file for WRP data. In addition, it causes a new WRP record to be displayed on the WLM Resource Pool page of the z/OSMF Workload Management task. The actual provisioning for the report class, and the definition and installation of the service definition, occurs only when the WRP definition is completed by the WLM administrator in z/OSMF.

On successful completion of a prime a WLM resource pool request, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 747](#).

For a description of the response content, see [“Response content” on page 748](#).

Request content

The request content is expected to contain a JSON object. See [Table 398 on page 748](#) for a description of the fields.

Table 398. Request content for the prime WLM resource pool request		
Field name	Required or Optional	Description
cloud-info	Required	<p>Specifies the attributes of the cloud:</p> <p>domain-name Name of the domain of systems</p> <p>domain-id Generated identifier of the domain of systems</p> <p>tenant-name Name of the tenant for the domain</p> <p>tenant-id Generated identifier of the tenant</p> <p>template-name Name of the software services template</p> <p>rdp-id Generated identifier of the resource pool</p>
wrp-data	Required	<p>Specifies the attributes of the WLM resource pool:</p> <p>wrp-name Name of the WLM resource pool</p> <p>service-level-agreements Array of service-level agreements, in the form "sla-name":"level", for example:</p> <pre> "service-level-agreements": [{"sla-name": "GOLD"}, {"sla-name": "SILVER"}] </pre> <p>report-class-name Name of the report class</p>

Authorization requirements

See [“Required authorizations”](#) on page 746.

HTTP status codes

For a successful request, the response body is provided, as described in [“Request content”](#) on page 747.

For a list of status codes, see [“Error handling”](#) on page 748.

Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 399 on page 748](#).

Table 399. Response content for a successful prime WLM resource pool request	
Field name	Description
status	Status of the request.
return-code	Return code of the request.
message	Message issued for the request.

Table 399. Response content for a successful prime WLM resource pool request (continued)	
Field name	Description
wrp-id	Identifier of the WLM resource pool
state	State of the request.

Example HTTP interaction

1. The example in Figure 336 on page 749 shows a request to prime a WLM resource pool.

```
POST https://host:port/zosmf/zwlm/rest/wrps
{
  "cloud-info":{
    "domain-name":"DOMAIN1",
    "domain-id":"12345ABC",
    "tenant-name":"Joey",
    "tenant-id":"IZU$000",
    "template-name":"CICSBasic",
    "rdp-id":"rdp01"
  }
  "wrp-data"{
    "wrp-name":"WRP1",
    "service-level-agreements":[
      {"sla-name":"GOLD"},
      {"sla-name":"SILVER"}
    ],
    "report-class-name":"Joey00"
  }
}
```

Figure 336. Sample request to issue a prime WRP request

The following is the response body for the request.

```
{
  "status":"success",
  "return-code":"0",
  "message":null,
  "wrp-id":"1090f34e-0a5a-4506-b553-91e932a46f3e",
  "wrp-name":"WRP1",
  "state":"initialized"
}
```

Figure 337. Sample response body

Delete a WLM resource pool

Use this operation to delete the record for a WLM resource pool.

HTTP method and URI path

```
DELETE /zosmf/zwlm/rest/wrps/wrpid
```

In this request, *wrpid* is the identifier of the WLM resource pool.

Query parameters

None.

Description

This operation:

- Removes a WLM resource pool record from the persistent file for WLM resource pools
- Deprovisions the report class in the current installed service definition if it is not referenced by any other classification rule
- Deletes the WLM resource pool record from the WLM Resource Pool page of the z/OSMF Workload Management task.

On successful completion, a response body that describes the request is returned.

For a description of the response content, see [“Response content” on page 750](#).

Request content

None.

Authorization requirements

See [“Required authorizations” on page 746](#).

HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 750](#).

For a list of status codes, see [“Error handling” on page 746](#).

Response content

On completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 400 on page 750](#).

Table 400. Response content for a delete WLM resource pool request	
Field name	Description
status	Status of the request.
return-code	Return code of the request.
message	Message issued for the request.

Example HTTP interaction

1. The example in [Figure 338 on page 750](#) shows a request to delete a WLM resource pool.

```
DELETE https://host:port/zosmf/zwlrm/rest/wrps/1090f34e-0a5a-4506-b553-91e932a46f3e
```

Figure 338. Sample request to issue a delete WLM resource pool request

The following is the response body for the successful request.

```
{
  "status": "success",
  "return-code": "0",
  "message": null
}
```

Figure 339. Sample response body

Construct a WLM service definition

Use this operation to construct a new service definition based on the current installed service definition.

HTTP method and URI path

PUT /zosmf/zwlrm/rest/policy/inspolicy

Query parameters

None.

Description

This operation constructs a new service definition based on the current installed service definition. A description for the report class is generated based on the domain name and domain ID.

On successful completion, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 751](#).

For a description of the response content, see [“Response content” on page 752](#).

Request content

The request content is expected to contain a JSON object. See [Table 401 on page 751](#) for a description of the fields.

Table 401. Request content for the construct a WLM service definition request		
Field name	Required or Optional	Description
cloud-info	Required	Specifies attributes of the cloud: wrp-id Identifier of the WLM resource pool.
provision-data	Required	Specifies the attributes of the WLM service definition: classification-rules Array of attributes, in the form "attribute-name":"value", for example: <pre>"classification-rules": [{"service-level-agreement":"GOLD", "qualifier-value":"CICSL00"}]</pre>

Authorization requirements

See [“Required authorizations” on page 746](#).

HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 751](#).

For a list of status codes, see [“Error handling” on page 746](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 402 on page 752](#).

Table 402. Response content for a successful construct a WLM service definition request	
Field name	Description
status	Status of the request.
messages	Message issued for the request.
result	<p>Result for the request.</p> <p>classification-rules Array of attributes, in the form "attribute-name":"value", for example:</p> <pre>"classification-rules": [{ "classification-rule-id": "id", "service-class-name": "name", "report-class-name": "name" }]</pre>

Example HTTP interaction

1. The example in [Figure 340 on page 752](#) shows a request to construct a service definition based on the current installed definition.

```
PUT https://host:port/zosmf/zwlm/rest/policy/inspolicy
{
  "cloud-info": {
    "wrp-id": "e5697dd6-88da-43f8-8f89-
bbdf9537b296",
    }
  "provision-data": {
    "classification-rules": [
      {
        "service-level-agreement": "GOLD",
        "qualifier-value": "CICSL00"
      }
    ]
  }
}
```

Figure 340. Sample request to construct a service definition based on the current installed definition

The following is the response body for the request.

```
{
  "state": "success",
  "return-code": "0",
  "message": null,
  "result": {
    "classification-rules": [
      {
        "classification-rule-id": "1090f34e-0a5a-4506-b553-91e932a46f3e",
        "service-class-name": "DEFAULT",
        "report-class-name": "JOEY00"
      }
    ]
  }
}
```

Figure 341. Sample response body

Construct a WLM service definition with remove and install

Use this operation to construct a new service definition by removing the classification rule, then installing the new service definition.

HTTP method and URI path

```
PUT /zosmf/zwlm/rest/policy/inspolicy
```

Query parameters

None.

Description

This operation constructs a new service definition by removing the classification rule, then installing the new service definition.

On successful completion, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 753](#).

For a description of the response content, see [“Response content” on page 754](#).

Request content

The request content is expected to contain a JSON object. See [Table 403 on page 753](#) for a description of the fields.

Table 403. Request content for the construct a WLM service definition request		
Field name	Required or Optional	Description
cloud-info	Required	Specifies cloud-related attributes: wrp-id Identifier of the WLM resource pool
deprovision-data	Required	Specifies the attributes of the WLM service definition: classification-rules Array of attributes, in the form "attribute-name":"value", for example: <pre>"classification-rules": [{"classification-rule-id":"id"}]</pre>

Authorization requirements

See [“Required authorizations” on page 746](#).

HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 753](#).

For a list of status codes, see [“Error handling” on page 746](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 404 on page 754](#).

Table 404. Response content for a successful construct a WLM service definition request	
Field name	Description
state	State of the request.
messages	Message issued for the request.

Example HTTP interactions

1. The example in [Figure 342 on page 754](#) shows a request to construct a service definition by removing the classification rule, then installing the new service definition.

```
PUT https://host:port/zosmf/zwlm/rest/policy/inspolicy
{
  "cloud-info": {
    "wlp-id": "e5697dd6-88da-43f8-8f89-bbdf9537b296",
    "deprovision-data": {
      "classification-rules": [
        {
          "classification-rule-id": "1090f34e-0a5a-4506-b553-91e932a46f3e"
        }
      ]
    }
  }
}
```

Figure 342. Sample request to construct a service definition by removing the classification rule, then installing the new service definition

The following is the response body for the request.

```
{
  "state": "success",
  "message": null,
}
```

Figure 343. Sample response body

RMF metering services

The Resource Measurement Facility REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. This interface allows a client application to interact with z/OSMF.

[Table 405 on page 754](#) lists the operations that the RMF services provide.

Table 405. Operations provided through the RMF services	
Operation	HTTP method and URI path
“Get metered data” on page 755 https://ibmid.acrolinx.cloud	GET /zosmf/izur/rest/meterdata
“Get DDS server time data” on page 758	GET /zosmf/izur/rest/ddstime

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 893](#).

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 404 Not found

Requested resource does not exist.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

Error logging

Errors from the z/OSMF notifications services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Get metered data

Use this operation to retrieve your metered data.

HTTP method and URI path

```
GET /zosmf/izur/rest/meterdata
```

Query parameters

Table 406. Query parameters		
Parameter	Description	Rule
TRG	Tenant resource group name.	Must contain one to eight characters, starting with an alphabetic character.
TRGLACS	Long-term average of CPU service (millions of service units) consumed by a tenant resource group. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGCP	Service units on general purpose processors consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.

Table 406. Query parameters (continued)

Parameter	Description	Rule
TRGAAP	Service units on zAAPs consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGIIP	Service units on zIIPs consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGCPN	General purpose processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGAAPN	zAAP processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGIIPN	zIIP processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
date	Specifies the start and end date/time of the RMF reporting interval.	Must be in the format of: startDate: yyyyymmdd endDate: yyyyymmdd
timeofday	The start time and end time based on the date of the operation to retrieve your metered data.	Must be in the format of: startTime: hhmm endTime: hhmm
duration	The length of the RMF reporting interval.	Must be in the format of: duration: hhmm

Description

This operation retrieves your metered data.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in metered data being retrieved, and a response body is returned. See [Figure 345 on page 758](#).

Request content

None.

Authorization requirements

Provide authorization level. For example, "User" or "Superuser". If there is no authorization level, specify "None".

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 407. HTTP error response codes for a get metered data request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested metered data does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

Example HTTP interaction

In [Figure 344 on page 757](#), a request is submitted to retrieve your metered data.

```
GET https://zosmfhost:1035/zosmf/izur/rest/meterdata?TRG=TRG1&TRGLACS=TRGLACS1&TRGCP=TRGCP1&TRGAAP=TRGAAP1&TRGIIP=TRGIIP1&TRGCPN=TRGCPN1&TRGAAPN=TRGAAPN1&TRGIIPN=TRGIIPN1&date=20170724,20170724&timeofday=0500,2400&duration=0100
```

Figure 344. Sample request to get metered data

The following [Figure 345 on page 758](#), is the response body for the example GET metered data request.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="include/ddsm1-pp.xsl"?>
<ddsm1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="include/ddsm1.xsd">
  <server>
    <name>RMF-DDS-Server</name>
    <version>Z0SV2R2</version>
    <functionality>3206</functionality>
    <platform>z/OS</platform>
  </server>
  <postprocessor><metric id="0VW"><description>RMF Overview Report</description>
    <type>Interval</type></metric><version><smf-data>z/OS V2R2</smf-data><rmf-report>
    z/OS V2R2</rmf-report></version><resource><resname>TRX1</resname><restype>SYSTEM</restype>
  </resource><time-data><display-start locale="en-us">07/31/2017-09.34.41</display-start>
    <display-end locale="en-us">07/31/2017-09.44.34</display-end><report-interval
    unit="hours">000:09:51</report-interval><cycle unit="milliseconds">1000</cycle>
  </time-data>
  <segment id="1"><name>Overview Report</name>
    <part id="2">
      <var-list id="3"><var><name>Number of Intervals</name><value>1</value></var><var>
      <name>Total Length of Intervals</name><value>00.09.51</value></var>
    </var-list>
    <table id="4">
      <column-headers><col type="T">Date (mm/dd)</col><col type="T">Time (hh.mm.ss)</col>
      <col type="T">Interval Length (hh.mm.ss)</col><col condition="TRGCP" qualifier="TRGRMFN1"
      type="N">TRGCP1</col>
      <col condition="TRGLACS" qualifier="TRGRMFN1" type="N">TRGLACS1</col><col condition="TRGAAP"
      qualifier="TRGRMFN1" type="N">TRGAAP1</col><col condition="TRGIIP" qualifier="TRGRMFN1"
      type="N">TRGIIP1</col></column-headers>
      <row refno="1"><col>07/31</col><col>09.34.41</col><col>00.09.51</col><col>0</col><col>0</col>
      <col>0</col></row>
    </table></part></segment>
  </postprocessor>
  <postprocessor><metric id="0VW"><description>RMF Overview Report</description><type>Interval</type>
    </metric><version><smf-data>z/OS V2R2</smf-data><rmf-report>z/OS V2R2</rmf-report></version><resource>
    <resname>T2</resname><restype>SYSTEM</restype>
  </resource><time-data><display-start locale="en-us">07/31/2017-09.35.06</display-start>
    <display-end locale="en-us">07/31/2017-09.44.34</display-end><report-interval unit="hours">000:09:26
    </report-interval><cycle unit="milliseconds">1000</cycle>
  </time-data>
  <segment id="1"><name>Overview Report</name>
    <part id="2">
      <var-list id="3"><var><name>Number of Intervals</name><value>1</value></var><var>
      <name>Total Length of Intervals</name><value>00.09.26</value></var>
    </var-list>
    <table id="4">
      <column-headers><col type="T">Date (mm/dd)</col><col type="T">Time (hh.mm.ss)</col>
      <col type="T">Interval Length (hh.mm.ss)</col><col condition="TRGCP" qualifier="TRGRMFN1"
      type="N">TRGCP1</col>
      <col condition="TRGLACS" qualifier="TRGRMFN1" type="N">TRGLACS1</col><col condition="TRGAAP"
      qualifier="TRGRMFN1" type="N">TRGAAP1</col><col condition="TRGIIP" qualifier="TRGRMFN1"
      type="N">TRGIIP1</col></column-headers>
      <row refno="1"><col>07/31</col><col>09.35.06</col><col>00.09.26</col><col>0</col><col>0</col>
      <col>0</col></row>
    </table></part></segment>
  </postprocessor>
</ddsm1>

```

Figure 345. Sample response

Get DDS server time data

Use this operation to retrieve your DDS server time data.

HTTP method and URI path

```
GET /zosmf/izur/rest/ddstime
```

Description

This operation retrieves your DDS server time data.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a DDS server time being retrieved, and a response body is returned. See [Figure 347 on page 760](#).

Request content

None.

Authorization requirements

Provide authorization level. For example, "User" or "Superuser". For no authorization level, specify "None".

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 408. HTTP error response codes for a get DDS server time data request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested DDS server time does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Response content

Example HTTP interaction

In [Figure 346](#) on [page 759](#), a request is submitted to retrieve DDS server time data.

```
GET https://zosmfhost:1035/zosmf/izur/rest/ddstime
```

Figure 346. Sample request to get DDS server time data

The following [Figure 347](#) on [page 760](#), is the response body for the example GET DDS server time data.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="include/ddsm1-m3.xsl"?>
<ddsm1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="include/ddsm1.xsd">
  <server>
    <name>RMF-DDS-Server</name>
    <version>Z0SV2R3</version>
    <functionality>3211</functionality>
    <platform>z/OS</platform>
  </server>
  <report>
    <metric id="CFOVER">
      <description>CFOVER (Coupling Facility Overview)</description>
      <format>report</format>
      <numcols>33</numcols>
    </metric>
    <resource>
      <reslabel>,PLEX1,SYSPLEX</reslabel>
      <restype>SYSPLEX</restype>
      <reslabelurl>%2CPLEX1%2CSYSPLEX</reslabelurl>
    </resource>
    <time-data>
      <local-start>20170927030820</local-start>
      <local-end>20170927031000</local-end>
      <utc-start>20170927070820</utc-start>
      <utc-end>20170927071000</utc-end>
      <local-prev>20170927030730</local-prev>
      <local-next>20170927031050</local-next>
      <display-start locale="en-us">09/27/2017 03:08:20</display-start>
      <display-end locale="en-us">09/27/2017 03:10:00</display-end>
      <gatherer-interval unit="seconds">100</gatherer-interval>
      <data-range unit="seconds">100</data-range>
    </time-data>
    <caption><var><name>CFOHPNAM</name><value>CTTEST1</value></var><var><name>CFOHPACD</name><value>03/08/01</value>
    </var><var><name>CFOHPACT</name><value>19.13.33</value></var><var><name>CFOHPREF</name><value>N0</value></var></caption>
    <row refno="1"><col>TESTCFN</col> <col>MDEV</col> <col>001</col> <col>8</col> <col>N/A</col> <col>0kay</col>
    <col>YES</col> <col>0.0</col> <col>1</col> <col>N/A</col> <col>N/A</col> <col>N/A</col> <col>1.0</col> <col>2.5</col>
    <col>6815744</col> <col>0</col> <col>100</col> <col>6815744</col> <col>0</col> <col>262144</col> <col>0</col> <col>1</col>
    <col>4</col> <col>0</col> <col>0FF</col> <col>0FF</col> <col>0</col> <col>0</col> <col>0.0</col> <col>0</col> <col>0</col>
    <col>0.0</col> <col>0</col></row>
    <column-headers> <col type="T">CFOPNAM</col> <col type="T">CFOPMOD</col> <col type="N">CFOPVER</col> <col type="N">CFOPLVL
    </col> <col type="T">CFOPDYND</col> <col type="T">CFOPSTAT</col> <col type="T">CFOPVOL</col> <col type="N">CFOPUTIP</col>
    <col type="N">CFOPDEF</col> <col type="T">CFOPPEDE</col> <col type="T">CFOPPSHR</col> <col type="T">CFOPPGT</col>
    <col type="N">CFOPPEFF</col> <col type="N">CFOPREQR</col> <col type="N">CFOPTSD</col> <col type="N">CFOPTSF</col>
    <col type="N">CFOPUTIS</col> <col type="N">CFOPTCS</col> <col type="N">CFOPDTS</col>
    <col type="N">CFOPDTUS</col> <col type="N">CFOPSYSC</col> <col type="N">CFOPSTCI</col> <col type="N">CFOPSTCO</col>
    <col type="T">CFOPMNT</col> <col type="T">CFOPRCV</col> <col type="N">CFOPSCMS</col> <col type="N">CFOPSCMA</col>
    <col type="N">CFOPSCMU</col> <col type="N">CFOPAUGS</col> <col type="N">CFOPAUGA</col> <col type="N">CFOPAUGU</col>
    <col type="N">CFOPMSC</col> </column-headers>
  </report>
</ddsm1>

```

Figure 347. Sample response

z/OS Compliance REST Interface

The z/OS compliance REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with the collection of security compliance evidence on a z/OS system, as described in this topic.

The Table 409 on page 760 lists the operations that the z/OS compliance REST interface services provide.

Table 409. Operations that are provided through the z/OS compliance REST interface services	
Operation	HTTP method and URI path
POST the request for compliance facts	POST zosmf/compliance/rest/v1/facts

Use the Swagger interface

You can use the Swagger interface to display information about the z/OS compliance REST APIs. The Swagger interface includes one section: Compliance APIs. For more information, see [“Using the Swagger interface”](#) on page 1.

Processing overview

This API is used to request the collection of compliance data from in-scope systems. In response, selected products and components collect and write compliance data to the SMF 1154 record type. The z/OS compliance REST interface services can be invoked by any HTTP client application that runs on the z/OS local system or a remote system. Your program (the client) initiates an HTTP request to the z/OS compliance REST interface. If the interface determines that the request is valid, it performs the requested service. After it performs the service, the z/OS compliance REST interface creates an HTTP response. If the request is successful, this response takes the form of an HTTP 2nn response. If the request is not successful, the response consists of a non-OK HTTP response code with details of the error that is provided in the form of a TEXT object.

Resource URLs

The URLs of the z/OS compliance REST interface have the format that is shown in [“#unique_34/unique_34_Connect_42_FormatResourceURLs” on page 761](#).

```
https://{host}:{port} /zosmf/compliance/rest/v1
```

Where:

- `https://{host}:{port}` specifies the target system address and port.
- `/zosmf/compliance/rest/v1` identifies the z/OS compliance REST interface.

HTTP Methods

The z/OS compliance REST interface provides the following HTTP methods: POST
POST the request for compliance facts.

Supported HTTP versions

z/OS compliance REST interface supports requests in either of the following protocols: HTTP/1.0 or HTTP/1.1.

Content Types

The data that is sent or returned by the HTTP methods has one of the following content types:

- JSON Content-Type: `application/json` is used for sent body data. For the detailed format of each JSON object, see the description for each operation.
- Plain text Content-Type: `plain/text`.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 201 Created

The request is successful. As a result, a resource is created.

HTTP 202 Accepted

The request is received and is accepted for processing. However, the processing is not yet complete.

HTTP 400 Bad request

Request that contains incorrect parameters.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the z/OS compliance REST interface services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

POST the request for compliance facts

Use this operation to post the request for compliance facts on z/OS.

HTTP method and URI path

```
POST /zosmf/compliance/rest/v1/facts
```

Where:

- `/zosmf/compliance/rest/v1/` identifies the z/OS compliance REST interface.

Standard Headers

Use the following standard HTTP header with this request: Content-Type: application/json.

Custom Headers

None.

Request Content

Your request must include a JSON object that describes the objects to be posted. [Table 410 on page 762](#) lists the supported parameters.

Table 410. Supported input parameters for the z/OS compliance REST service		
Parameter	Required	Description
requestid	Yes	requestid identifies the request for collecting data. Specify a value of 1 to 16 characters. Note: The requestid is saved in each SMF1154 record and helps identify the SMF 1154 data written for the request. Hence the requestid should be as unique as possible for each invocation of the API.
systemName	Yes	systemName is a list of system names for which compliance data is to be collected. Each system name is 8 characters. Up to 32 systems per sysplex can be specified.

Example HTTP interaction

The [Figure 348](#) on [page 763](#) shows an example of submitting a request to show the details of compliance on system pev051.

```
POST /zosmf/compliance/rest/v1/facts HTTP/1.1
Host: pev051.pok.ibm.com

{
  "requestid": "123e456f78a9d0bc",
  "systemname": ["SY0", "SY1"]
}
```

Figure 348. Sample request to post the request for compliance

The [Figure 349](#) on [page 763](#) shows the response body for the example POST the request for compliance facts.

```
HTTP/1.1 200 OK
```

Figure 349. Sample response from post the request for compliance

z/OS console services

The z/OS console services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. The z/OS console services provide a programming interface for performing z/OS console operations.

With the z/OS console services, you can issue system commands and work with both solicited messages (messages that were issued in response to the command) and unsolicited messages (other messages that might or might not have been issued in response to the command). z/OS console services establish an extended MCS (EMCS) console, which is then used to issue commands and receive messages.

[Table 411](#) on [page 763](#) lists the operations that the z/OS console services provide.

Table 411. Operations provided through the z/OS console services.	
Operation	HTTP method and URI path
“Issue a command from a system console” on page 765	PUT /zosmf/restconsoles/consoles/ <i>consolename</i> PUT /zosmf/restconsoles/consoles/defcn
“Get a command response” on page 782	GET /zosmf/restconsoles/consoles/ <i>console-name</i> /solmsgs/ <i>Ckey-number</i> GET /zosmf/restconsoles/consoles/defcn/solmsgs/ <i>Ckey-number</i>
“Get the detect result for unsolicited messages” on page 788	GET /zosmf/restconsoles/consoles/ <i>consolename</i> /detections/ <i>Dkey-number</i> GET /zosmf/restconsoles/consoles/defcn/detections/ <i>Dkey-number</i>
“Get messages from a hardcopy log” on page 795	GET /zosmf/restconsoles/v1/log

Using the Swagger interface

You can use the Swagger interface to display information about the z/OS console services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Required authorizations

Your user ID must have the same authority when issuing a command with the z/OS console services as when issuing a command through a console on a z/OS system.

The required authority is:

- READ access to the MVS.MCSOPER.*console* resource in the OPERCMDS class, where *console* is the name of the EMCS console that is used to issue the command
- READ access to the CONSOLE resource in the TSOAUTH class
- READ access to the <SAF_PREFIX>.*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF_PREFIX>.IzuManagementFacilityRestConsoles.izuUsers resource name in the EJBROLE class.

If you want to get messages from logs, the following is the required authority:

- READ access to the SYSPLEX.OPERLOG resource in the LOGSTRM class
- READ access to the <sysname>.+MASTER+.SYSLOG.*.* resource in the JESSPOOL class, where <sysname> is the name of the system on which the SYSLOG resides.

z/OS console services use z/OSMF TSO/E address space services to create a TSO address space as the host for an EMCS console. To use TSO/E address space services, you must have:

- READ access to resource *account* in class ACCTNUM, where *account* is the value specified in the COMMON_TSO ACCT option in parmlib
- READ access to resource CEA.CEATSO.TSOREQUEST in class SERVAUTH
- READ access to resource *proc* in class TSOPROC, where *proc* is the value specified with the COMMON_TSO PROC option in parmlib.

You must also ensure that the z/OSMF started task user ID, which is IZUSVR by default, has READ access to resource CEA.CEATSO.TSOREQUEST in class SERVAUTH.

The TSO/E address space services authority might already be defined if you are using z/OS data set and file REST services, as those services require similar authority.

Configuration

z/OS console services use the TSO CONSOLE command to establish an EMCS console, which allows you to issue system commands and retrieve the messages that are issued in response. Console attributes such as ROUTCODE and AUTH affect the messages that the EMCS console can receive and the commands that the console can issue. When you use the z/OS console services, be sure that the EMCS console that is established has the desired attributes. For information, see [Extended MCS consoles in z/OS MVS Planning: Operations](#).

In addition, be aware that messages can be suppressed due to settings in the active MPFLSTxx member of parmlib. If a message associated with a command response is suppressed, a REST API call that attempts to detect that message will fail.

To control the parameters that z/OS console services use when creating a TSO address space as the host for an EMCS console, use parmlib option COMMON_TSO ACCT(IZUACCT) REGION(50000) PROC(IZUFPROC). Configure this setting before z/OS console services are to be used. Otherwise, default values are used with z/OS console services.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. The HTTP status codes are described in the topics for the individual services.

In addition, a JSON object describes the error.

Error logging

Errors from the z/OS console services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Issue a command from a system console

Use this operation to issue a command by using a system console.

HTTP method and URI path

```
PUT /zosmf/restconsoles/consoles/consolename
PUT /zosmf/restconsoles/consoles/defcn
```

Where:

consolename

Is the name of the EMCS console that is used to issue the command. The name must be 2 - 8 characters long, and cannot be defcn, which is reserved.

defcn

Indicates that the name of the console that is used to issue the command is generated by the REST Console API, by adding CN to the logon user ID. For example, if the logon user ID is CJOEY, the console name is CJOEYCN. If the user ID is longer than 6 characters, the user ID is truncated. For example, if the user ID is ZOSMFAD, the console name is ZOSMFACN.

Query parameters

None.

Description

This operation issues a command, based on the properties that are specified in the request body. On successful completion, HTTP status code 200 is returned. A JSON object typically contains the command response.

When a command is issued synchronously, the console API attempts to get the solicited messages immediately after the command is issued. If there are no messages available within a certain time interval, approximately 3 seconds when your system workload is not high, the API returns "cmd-response": "" in the response body.

A value for cmd-response of the empty string, "", usually means that there is no command response. However, it is also possible that the command response arrived after 3 seconds. If that is the case, you can use the cmd-response-url field in the response body to retrieve the command response. You might do this several times to ensure that all messages that are related to the command are retrieved.

Alternatively, you might examine unsolicited messages, that is, additional messages that are not part of the command response. To do so, you can issue the command with option unso1 - key to detect a keyword in the unsolicited messages.

For the properties that you can specify on the request body, see [“Request content”](#) on page 766.

For a description of the response content, see [“Response content”](#) on page 777.

The request body can include properties that you can use to control console attributes, such as 'auth', 'routcode', 'mscope', 'storage', and 'auto'. Alternatively, you can use System Authorization Facility or SAF to control console attributes. The RACF ADDUSER command with the OPERPARM parameter sets console attributes when a user establishes an EMCS console. Using the ADDUSER command to control console attributes requires that you know in advance the name of the EMCS console that the z/OS Console service

uses. The console name is either a name that you specify on the Issue Command service, or a name that the service generates, as described in “Issue a command from a system console” on page 765. For example, if user CJOEY plans to accept the default console name, CJOEYCN, this user can issue this RACF command to set console attributes for the console:

```
ADDUSER CJOEYCN OPERPARM(AUTH(MASTER) ROUTCODE(ALL))
```

Note: To use the properties 'auth', 'routcode', 'mscope', 'storage', or 'auto' in your request, your user ID requires at least READ access to resource CONOPER in class TSOAUTH.

Request content

The request content is expected to contain a JSON object. See [Table 412 on page 766](#) for a description of the fields.

Table 412. Request content for the issue command request		
Field name	Required or Optional	Description
cmd	Required	Specifies the command to issue.
sol-key	Optional	<p>Specifies a keyword that you want to detect in solicited messages, that is, the command response. Case is not significant.</p> <p>This value can be a string or a regular expression. To use a regular expression, you must also set the solKeyReg property.</p> <p>The regular expression can only contain one of the following special regex quantifiers: (*, +, {n}, {n,}, {n,m}).</p>
unsol-key	Optional	<p>Specifies a keyword that you want to detect in unsolicited messages. Case is not significant.</p> <p>This value can be a string or a regular expression. To use a regular expression, you must also set the unsolKeyReg property.</p> <p>The regular expression can only contain one of the following special regex quantifiers: (*, +, {n}, {n,}, {n,m}).</p> <p>Message suppression can prevent the return of an unsolicited message. To determine whether a particular message ID is suppressed through the message processing facility on your system, enter the following command to list the MPF settings: D MPF.</p>
detect-time	Optional	Indicates how long the console attempts to detect the value of unsol-key in the unsolicited messages. The unit is seconds. For example, if the value of detect-time is 10, the console checks the unsolicited messages for 10 seconds. The default is 30 seconds.
async	Optional	<p>Indicates the method of issuing the command:</p> <p>Y Asynchronously</p> <p>N Synchronously.</p> <p>If you omit this property, N is used by default.</p>

Table 412. Request content for the issue command request (continued)

Field name	Required or Optional	Description
system	Optional	Name of the system in the same sysplex that the command is routed to. The default is the local system.
unsol-detect-sync	Optional	<p>Indicates how to detect the keyword that is specified with the unsol-key field from unsolicited messages:</p> <p>Y</p> <p>Synchronously detect the keyword from unsolicited messages. The request is not returned until the unsol-detect-timeout value elapses or the detection result is complete.</p> <p>N</p> <p>Asynchronously detect the keyword from unsolicited messages. The request is returned immediately with the detection-url. The client application must invoke the value of detection-url to poll the result of the detection asynchronously.</p> <p>If you omit this property, N is used by default.</p>
unsol-detect-timeout	Optional	Indicates how long, in seconds, the request is blocked when the value for unsol-detect-sync is Y and the detection result is not complete. The default value, 20 seconds, is used when this field is not specified and the value for unsol-detect-sync is Y.
solKeyReg	Optional	<p>If the sol-key property is specified, this property indicates whether sol-key represents a regular expression.</p> <p>Y</p> <p>sol-key is a regular expression.</p> <p>N</p> <p>sol-key is a normal string.</p> <p>For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key:</p> <pre>a[\s]+regular</pre> <p>If you omit this property, N is used by default.</p>

Table 412. Request content for the issue command request (continued)

Field name	Required or Optional	Description
unsolKeyReg	Optional	<p>If the unsol-key property is specified, this property indicates whether unsol-key represents a regular expression.</p> <p>Y unsol-key is a regular expression.</p> <p>N unsol-key is a normal string.</p> <p>For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key:</p> <pre>a[\s]+regular</pre> <p>If you omit this property, N is used by default.</p>
auth	Optional	<p>Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p>MASTER Allows this console to act as a master console, which can issue all MVS operator commands.</p> <p>ALL Allows this console to issue system control commands, input/output commands, console control commands, and informational commands.</p> <p>INFO Allows this console to issue informational commands.</p> <p>CONS Allows this console to issue console control and informational commands.</p> <p>IO Allows this console to issue input/output and informational commands.</p> <p>SYS Allows this console to issue system control commands and informational commands.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>

Table 412. Request content for the issue command request (continued)

Field name	Required or Optional	Description
routcode	Optional	<p>Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p>ALL All routing codes.</p> <p>NONE No routing codes.</p> <p>(routing-codes) One or more routing codes or sequences of routing codes. The routing codes can be list of n and n1:n2, where n, n1, and n2 are integers 1 - 128, and n2 is greater than n1.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>
mscope	Optional	<p>The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p>(system-name) List of one or more system names, where system-name can be any combination of A - Z, 0 - 9, # (X'7B'), \$ (X'5B'), or @ (X'7C').</p> <p>LOCAL System on which the console is active.</p> <p>ALL All systems.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>
storage	Optional	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for message queuing to this console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p> <p>If the expected console message size is large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>

Table 412. Request content for the issue command request (continued)

Field name	Required or Optional	Description
auto	Optional	<p>Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p>YES The console receives messages that are eligible for automation.</p> <p>NO The console does not receive messages that are eligible for automation.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>

Authorization requirements

See [“Required authorizations”](#) on page 764.

HTTP status codes

For a successful request, HTTP status code 200 is returned and the response body is provided, as described in [“Request content”](#) on page 766.

For unsuccessful requests, the service returns the status codes that are described in [Table 413](#) on page 770.

Table 413. HTTP error response codes for an issue command request

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	3	No match for method PUT and pathInfo=' %s '.	<p>The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL.</p> <p>A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.</p>
400	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
400	1	6	The Content-Type ' %s ' cannot be handled, 'application/json' is expected.	The Content-Type, %s, in the original request contains an incorrect value for the HTTP Content-Type header. The z/OS Console API accepts only application/json for the Content-Type. Update the value of the HTTP Content-Type header and make sure that the request body is in JSON format.

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	11	Format of parameter 'rsize' is wrong, it cannot be changed to a number.	The rsize parameter requires a numeric value, but the supplied value is not a number. Change the value to a number.
400	1	12	The body of the request is not in JSON format.	The request body must be in JSON format, but the supplied request body is not in JSON format. Correct the request body to be in JSON format.
400	1	13	Cannot find 'cmd' in request body, or value of 'cmd' is empty. No command is issued.	No cmd field was found in the request body, or the cmd field is empty. The cmd field specifies the command to be issued. No command is issued. Ensure that the request body includes a cmd field with a value.
400	1	14	Incorrect console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.
400	1	17	Command length must be less than 127.	The value of the cmd field exceeds the maximum length of a command, which is 126 characters. Provide a valid command.
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> • Verify that the logon procedure exists and is valid. • Specify a different region size, or use the installation-defined default. • If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session. • If you want to use profile sharing, do the following: <ul style="list-style-type: none"> – Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR. – Turn on profile sharing in the user settings for the z/OSMF ISPF task. – For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option. • Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	25	Invalid value for <i>sol-key</i> or <i>unsol-key</i> . The regular expression should only contain one of the following special regex quantifiers: (*, +, {n}, {n,m}).	If regular expressions in <i>sol-key</i> and <i>unsol-key</i> are used, make sure the regular expressions only contain one of the following special regex quantifiers: (*, +, {n}, {n,m}).
500	2	7	Internal https connection timeout.	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	8	I/O error when connecting TSO service	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned non-200 status code when creating TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.
500	3	3	REST TSO service returned non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data that is returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	6	Unknown error occurred when creating or getting the TSO AS.	An unknown error occurred during an attempt to create a TSO address space. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces that are allowed for the current user has been reached.	Refer to message IZUG1127E for a detailed explanation. Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console. To cancel a TSO/E address space, issue the <code>C u=user-ID,a=ASID</code> command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	14	An exception occurred when connecting to the TSO/E address space. Error description: %s.	The request failed because an error occurred. The context of the error is provided in the message text: <code>error description, %s</code> . For details about the error, check the z/OSMF logs. Correct any errors that you find. If the problem persists, contact IBM Support and provide the error details.
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation. Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console. To cancel a TSO/E address space, enter the command <code>C u=user-ID,a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	19	The TSO/E user account number for the console is not defined for use.	Refer to message IKJ56486I for a detailed explanation. One of the following occurred: <ul style="list-style-type: none"> The specified account number is not defined to the RACF database. The RACF administrator must first define the account number as a RACF resource and then give the user access using the PERMIT command. However, if the procedure is not in the procedure library, the logon attempt will continue to fail. <ul style="list-style-type: none"> The RACF class ACCTNUM is not active. The RACF administrator must activate the RACF class ACCTNUM using the SETROPTS command.
500	3	20	TSO/E user account number for the console has not been authorized for the user.	Refer to message IKJ56487I for a detailed explanation. The specified account number is defined to the RACF database. However, this particular user ID is not allowed to use it.
500	3	21	TSO/E user account number for the console is not valid.	Refer to message IKJ56702I for a detailed explanation. The specified account number is incorrect.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Incorrect parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	No application data returned when initialize time zone for console service.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	4	Unexpected IEE136I message: %s	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Failed to retrieve the time zone from message IEE136I. Local Time: %s, UTC Time: %s	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	7	Create console failed due to a TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The number of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	10	The requested EMCS console exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation. Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization .

Table 413. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	17	The value for auth is not valid. The following values are valid: MASTER, ALL, INFO, CONS, IO, SYS.	The command authority for the console is not valid. Valid values are: <ul style="list-style-type: none"> • MASTER • ALL • INFO • CONS • IO • SYS
500	5	18	The value for routcode is not valid.	The routing codes for the console are not valid. Valid values are: <ul style="list-style-type: none"> • ALL • NONE • List of one or more routing codes in the range 1-128, enclosed in parentheses. Multiple entries are separated by commas.
500	5	19	The value for mscope is not valid.	The message scope for the console is not valid. Valid values are: <ul style="list-style-type: none"> • ALL • LOCAL • List of one or more system names, which are enclosed in parentheses. Multiple entries are separated by commas.
500	5	20	The value for storage is not valid.	The value for storage must be a number in the range 1 - 2000.
500	5	21	The value for auto is not valid.	The message automation status for the console is not valid. Valid values are: <ul style="list-style-type: none"> • YES • NO
500	5	22	The EMCS console cannot be activated because the user has insufficient access to resource profile CONOPER.	User is not authorized to CONOPER. The user requires at least READ access for resource profile CONOPER CLASS(TSOAUTH).
500	5	23	OPERPARM value is not valid.	Check the console's OPERPARM values for syntax errors. Correct the errors and try the request again.

Table 413. HTTP error response codes for an issue command request (continued)				
HTTP Status	Return Code	Reason Code	Reason	Description
500	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
500	8	14	Cannot get the command response.	The z/OS Console API failed to get the command response. Try the request again. If the problem persists, contact the z/OSMF administrator.

Response content

On successful completion, the service returns a response body, which contains a JSON object. The JSON object varies depending on whether the request was synchronous or asynchronous. For a description of fields in the JSON object, see either of the following tables:

- [Table 414 on page 777](#)
- [Table 415 on page 778](#).

Table 414 on [page 777](#) describes the response content for a successful synchronous issue command response.

Table 414. Response content for a successful synchronous issue command request	
Field name	Description
cmd-response	Command response.
cmd-response-url	URL that can be used to retrieve the command response later when the value for cmd-response is empty.
cmd-response-uri	URI that can be used to retrieve the command response later when the value for cmd-response is empty. The URI starts with /zosmf.
cmd-response-key	Key that can be used to retrieve the command response later when the value for cmd-response is empty.
consoleAuth	Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
consoleRoutcde	Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
consoleMscope	The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.

Table 414. Response content for a successful synchronous issue command request (continued)

Field name	Description
consoleStorage	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for queuing messages to this console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p> <p>Note: If the operand OPERPARM(STORAGE) is not specified, the console uses STORAGE(1) when the session is established.</p> <p>If the expected console message size is very large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>
consoleAuto	<p>Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p>
sol-key-detected	<p>Returned when sol-key was specified, and unsol-detect-sync was specified as N or not specified. If the keyword was detected in the command response, the value is true. Otherwise, the value is false.</p>
ipcmmsgqbytes	<p>Maximum number of bytes in a single message queue. The returned value is a decimal value in the range 0 - 2147483647. The default value is 2147483647 (or 2G).</p> <p>This field is included only for the first command to be issued with this console. On subsequent uses, this field is omitted.</p> <p>Note: If this value is less than the maximum (2147483647), the z/OSMF Operator Consoles task might lose some solicited or unsolicited messages. If so, your system programmer can use the SETOMVS or SET OMVS command to increase the value of IPCMSGQBYTES.</p>

Table 415 on page 778 describes the response content for a successful asynchronous issue command response.

Table 415. Response content for a successful asynchronous issue command request

Field name	Description
cmd-response-url	URL that can be used to retrieve the command response.
cmd-response-uri	URI that can be used to retrieve the command response. The URI starts with / zosmf.
cmd-response-key	Key that can be used to retrieve the command response.
consoleAuth	Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
consoleRoutcde	Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
consoleMscope	The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.

Table 415. Response content for a successful asynchronous issue command request (continued)

Field name	Description
consoleStorage	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for queuing messages to this console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p> <p>Note: If the operand OPERPARM (STORAGE) is not specified, the console uses STORAGE (1) when the session is established.</p> <p>If the expected console message size is very large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>
consoleAuto	Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
detection-url	The URL that can be used later to retrieve the detection result for detecting a keyword from unsolicited messages. Returned when unsol-key was specified to detect a keyword in unsolicited messages, and unsol-detect-sync was specified as N or not specified.
detection-uri	The URI that can be used later to retrieve the detection result for detecting a keyword from unsolicited messages. Returned when unsol-key was specified to detect a keyword in unsolicited messages, and unsol-detect-sync was specified as N or not specified.
detection-key	Returned when unsol-key was specified to detect a keyword in unsolicited messages. You can use this value to retrieve the result.
status	<p>Status of the unsolicited detection request. Returned when sol-key was specified, and unsol-detect-sync is specified as Y. The values are:</p> <p>expired The detection request is expired. No matching record in the unsolicited messages was found in the time that is specified by detect-time.</p> <p>detected Matching records in the unsolicited messages were found in the time that is specified by detect-time. msg contains the message that contains the keyword.</p> <p>timeout The unsol-detect-timeout elapsed before the detection result completed.</p> <p>detection-url The URL that can be used to retrieve the detection result for detecting a keyword from unsolicited messages.</p> <p>detection-uri The URI that can be used to retrieve the detection result for detecting a keyword from unsolicited messages.</p> <p>detection-key The key that can be used to retrieve the unsolicited keyword detection result.</p>

Table 415. Response content for a successful asynchronous issue command request (continued)

Field name	Description
ipcmmsgqbytes	<p>Maximum number of bytes in a single message queue. The returned value is a decimal value in the range 0 - 2147483647. The default value is 2147483647 (or 2G).</p> <p>This field is included only for the first command to be issued with this console. On subsequent uses, this field is omitted.</p> <p>Note: If this value is less than the maximum (2147483647), the z/OSMF Operator Consoles task might lose some solicited or unsolicited messages. If so, your system programmer can use the SETOMVS or SET OMVS command to increase the value of IPCMSGQBYTES.</p>

The client application can use any one of detection-url, detection-uri, or detection-key to retrieve the detection result.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 416. Response content for an unsuccessful issue command request

Field name	Description
return-code	Category of the error.
reason-code	Specific error.
reason	Text that describes the cause of the error.

Example HTTP interactions

1. The example in [Figure 350 on page 780](#) shows a request to issue the system command **d a, pegasus** synchronously.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/ibmusecn
{"cmd":"d a,PEGASUS", "routcode" : "ALL"}
```

Figure 350. Sample request to issue a command synchronously

The following is the response body for the request. In the response, \r is the return character.

```
{
  "cmd-response-key":"C6557643",
  "cmd-response-uri":"https://pev076.pok.ibm.com:443/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C6557643",
  "IPCMSGQBYTES":"2147483647",
  "consoleRoutcode":"ALL",
  "consoleMscope":"ALL",
  "consoleAuto":"NO",
  "cmd-response-uri":"/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C6557643",
  "cmd-response":"CNZ4106I 04.22.11 DISPLAY ACTIVITY 532\r
JOBS   M\ S   TS USERS  SYSAS  INITS  ACTIVE\ MAX VTAM   OAS\r 00002   00015   00002   00032   00005   00001\
00020 00011\r
PEGASUS PEGASUS *OMVSEX IN  SO  A=0038  PER=NO  SMC=000\r
PGN=N\ A  DMN=N\ A  AFF=NONE\r
CT=006.589S  ET=05.49.06\r
WUID=STC00061 USERID=ZOSMFAD\r
WKL=SYSTEM  SCL=SYSSTC  P=1\r
RGP=N\ A  SRVR=NO  QSC=NO\r
ADDR SPACE ASTE=01ED0E00",
  "consoleStorage":"1024",
  "consoleAuth":"INFO"}
```

Figure 351. Sample response body

2. The example in [Figure 352 on page 781](#) shows a request to issue the system command `d a,PEGASUS` synchronously, and attempt to detect PEGASUS in the command response.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/ibmusecn
{"cmd":"d a,PEGASUS","sol-key":"PEGASUS"}
```

Figure 352. Sample request to issue a command and detect a keyword

The following is the response body for the request.

```
{"cmd-response": "CNZ4106I 07.30.59 2016.011 DISPLAY ACTIVITY 070\tr JOBS M/S TS USERS SYSAS INITS ACTIVE/MAX
VTAM OAS\tr 00003 00013 00002 00032 00011 00001\V00020 00015\tr PEGASUS NOT FOUND",
"sol-key-detected":true,"cmd-response-uri":"\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-url":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-key":"C005291"}
```

Figure 353. Sample response body

3. The example in [Figure 354 on page 781](#) shows a request to issue the system command `s PEGASUS` asynchronously and attempt to detect PEGASUS in the unsolicited messages.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":"s PEGASUS","unsol-key":"PEGASUS","async":"Y"}
```

Figure 354. Sample request to issue a system command asynchronously

The following is the response body for the request.

```
{"cmd-response-uri":"\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"detection-uri":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\detections\dec6800",
"detection-key":"dec6800",
"cmd-response-url":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-key":"C005291"}
```

Figure 355. Sample response body

4. The example in [Figure 356 on page 781](#) shows a request to issue an `s PEGASUS` command synchronously, by using the default console, and detect keyword PEGASUS in the unsolicited messages synchronously. The keyword is found in unsolicited messages before the timeout is reached.

```
PUT https://PEV076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":"s PEGASUS","unsol-key":"PEGASUS","unsol-detect-sync":"Y"}
```

Figure 356. Sample request to issue an s PEGASUS command synchronously and detect the keyword PEGASUS

The following is the response body for the request.

```
{"status":"detected","cmd-response":"BPXM023I (ZOSMFAD) CFZ02300I: Configuration property
httpAuthType is not supported. Setting ignored.", "msg":"$HASP100 PEGASUS ON STCINRDR"}
```

Figure 357. Sample response body

5. The example in [Figure 356 on page 781](#) shows a request to issue an `s PEGASUS` command asynchronously, by using the default console, and detect keyword XIAOX in the unsolicited messages

synchronously. The detection result is not complete before the timeout (the default of 20 seconds) was reached.

```
PUT https://PEV076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":{"s PEGASUS","async":"Y","unsol-key":"XIAOX","unsol-detect-sync":"Y"}}
```

Figure 358. Sample request to issue an s PEGASUS command asynchronously and detect the keyword XIAOX

The following is the response body for the request.

```
{"cmd-response-uri":"/zosmf/restconsoles/consoles/defcn/solmsgs/C2790426","detection
-url":"https://PEV076.pok.ibm.com:443
/zosmf/restconsoles/consoles/defcn/detections/D5303564","detection-uri":"
/zosmf/restconsoles/consoles/defcn/detections
/D5303564","detection-key":"D5303564","status":"timeout","cmd-response-url":"https:
//PEV076.pok.ibm.com:443/zosmf/restconsoles/consoles/defcn/solmsgs/C2790426",
"cmd-response-key":"C2790426"}
```

Figure 359. Sample response body

Get a command response

Use this operation to get the response to a command that was issued asynchronously with the Issue Command service.

HTTP method and URI path

```
GET /zosmf/restconsoles/consoles/console-name/solmsgs/Ckey-number
GET /zosmf/restconsoles/consoles/defcn/solmsgs/Ckey-number
```

where:

consolename

is the name of the EMCS console that was used in the Issue Command request.

defcn

indicates that name of the console that was used to issue the command was generated by the REST Console API.

Ckey-number

is the command response key from the Issue Command request.

The URL is returned by the Issue Command request in the cmd-response-url field.

Query parameters

None.

Description

This operation gets the messages that were issued in response to a command that was issued asynchronously with the Issue Command service. For the properties that you can specify, see [“Request content” on page 783](#).

On successful completion, HTTP status code 200 is returned. The response content is described in [“Response content” on page 787](#).

The Issue Command service returns the URL of the command response in the cmd-response-url field. For more information about the response content of the Issue Command service, see [“Response content” on page 777](#).

Request content

None.

Authorization requirements

See [“Required authorizations” on page 764](#).

HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided, as described in [“Response content” on page 787](#).

Otherwise, the HTTP status codes in [Table 417 on page 783](#) are returned for the indicated errors.

<i>Table 417. HTTP error response codes for a get command response request</i>				
HTTP Status	Return Code	Reason Code	Reason	Description
400	1	3	No match for method GET and pathInfo=' %s '.	The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL. A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.
400	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
400	1	14	Invalid console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.

Table 417. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> • Verify that the logon procedure exists and is valid. • Specify a different region size, or use the installation-defined default. • If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session. • If you want to use profile sharing, do the following: <ul style="list-style-type: none"> – Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR. – Turn on profile sharing in the user settings for the z/OSMF ISPF task. – For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option. • Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned a non-200 status code when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.

Table 417. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	3	REST TSO service returned non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces allowed for the current user has been reached.	<p>Refer to message IZUG1127E for a detailed explanation.</p> <p>Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console.</p> <p>To cancel a TSO/E address space, issue the <code>C u=user-ID,a=ASID</code> command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.</p>

Table 417. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation. Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS, ALL</code> from the operator console. To cancel a TSO/E address space, enter the command <code>C u=user-ID, a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Invalid parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	4	Cannot retrieve local time zone.	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Cannot retrieve local time zone.	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 417. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	7	Create a console failed due to a TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The numbers of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	10	The requested EMCS console already exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation. Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization .
500	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
500	10	1	The message you requested cannot be retrieved due to earlier shutdown of z/OSMF server.	The z/OS Console API failed to get the command response. Try the request again. If the problem persists, contact the z/OSMF administrator.

Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 418](#) on page 787 lists the fields in the JSON object.

Table 418. Response content for a successful get command response request

Field name	Description
cmd-response	Command response
sol-key-detected	Returned when sol-key was specified on the Issue Command service. If the keyword specified with sol-key was found in the command response, the value is true. Otherwise, the value is false.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 419. Response content for an unsuccessful get command response request

Field name	Description
return-code	Category of the error.
reason-code	Specific error.
reason	Text that describes the cause of the error.

Example HTTP interaction

The example in [Figure 360 on page 788](#) shows a request to get the response to a system command that was issued asynchronously. The command was issued with a generated console name. The command response key returned by the issue command request is C003715.

```
GET https://pev061.pok.ibm.com:443/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C508135
```

Figure 360. Sample request to get the response for a system command that was issued asynchronously

The following is the response body for the request.

```
{ "cmd-response": "IEE215I 07.36.34 2016.011 PARMLIB DISPLAY 513\R PARMLIB DATA SETS SPECIFIED\R AT IPL\R ENTRY FLAGS
VOLUME DATA SET\R 1 S PEVTS3 CIMSSRE.R22ONLY.PARMLIB\r 2 S PEVTS3
CIMSSRE.R14ONLY.PARMLIB\r 3 S PEVTS3 CIMSSRE.R13ONLY.PARMLIB\r 4 S PEVTS3
CIMSSRE.R12ONLY.PARMLIB\r 5 S PEVTS3 CIMSSRE.PARMLIB\r 6 S PEVTS3 HDENNIS.ZOS17.PARMLIB\r
7 S CTTAK XESCT.PARMLIB\r 8 S CTTAK SYS1.PARMLIB\r 9 S SDR22 SYS1.PARMLIB.POK\r
S SDR22 SYS1.PARMLIB.INSTALL"} }
```

Figure 361. Sample response body for a get command response request

Get the detect result for unsolicited messages

Use this operation to get the result for detecting a keyword in unsolicited messages after an Issue Command request. The command must have been issued with the unsol-key field.

HTTP method and URI path

```
GET /zosmf/restconsoles/consoles/consolename/detections/Dkey-number
GET /zosmf/restconsoles/consoles/defcn/detections/Dkey-number
```

where:

consolename

is the name of the EMCS console that was used in the Issue Command request.

defcn

indicates that name of the console that was used to issue the command was generated by the REST Console API.

Dkey-number

is the detection key from the Issue Command request.

The URL is be returned by the Issue Command request in the detection-url field.

Query parameters

None.

Description

This operation gets the results of attempting to detect a keyword in the unsolicited messages that were issued following an Issue Command request. The keyword being detected was specified with the unsol-key field on the Issue Command service.

On successful completion, HTTP status code 200 is returned. The response content is described in [“Response content” on page 793](#).

The Issue Command service returns the URL in the detection-url field. For more information about the response content of the Issue Command service, see [“Response content” on page 777](#).

Request content

None.

Authorization requirements

See [“Required authorizations” on page 764](#).

HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided, as described in [“Response content” on page 793](#).

Otherwise, the HTTP status codes in [Table 420 on page 789](#) are returned for the indicated errors.

Table 420. HTTP error response codes for a detect result for unsolicited messages request				
HTTP Status	Return Code	Reason Code	Reason	Description
400	1	3	No match for method GET and pathInfo=' %s '.	The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL. A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.
400	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
400	1	14	Invalid console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.

Table 420. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> • Verify that the logon procedure exists and is valid. • Specify a different region size, or use the installation-defined default. • If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session. • If you want to use profile sharing, do the following: <ul style="list-style-type: none"> – Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR. – Turn on profile sharing in the user settings for the z/OSMF ISPF task. – For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option. • Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned a non-200 status code when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.

Table 420. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	3	REST TSO service returned a non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces allowed for the current user has been reached.	<p>Refer to message IZUG1127E for a detailed explanation.</p> <p>Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console.</p> <p>To cancel a TSO/E address space, issue the <code>C u=user-ID,a=ASID</code> command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.</p>

Table 420. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation. Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS, ALL</code> from the operator console. To cancel a TSO/E address space, enter the command <code>C u=user-ID, a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Invalid parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	4	Cannot retrieve local time zone.	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Cannot retrieve local time zone.	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 420. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	7	Create console failed due to TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The number of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	9	Cannot find the result for specified detection ID.	Cannot find the result for the specified detection ID. Ensure that the detection ID is correct.
500	5	10	The requested EMCS console already exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation. Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization .
500	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
500	10	2	The detection result you requested cannot be retrieved due to earlier shutdown of the z/OSMF server.	The detection result cannot be retrieved because of an earlier shutdown of z/OSMF server.

Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 421 on page 794](#) lists the fields in the JSON object.

Table 421. Response content for a successful get detect result request

Field name	Description
status	Status of the detection request: waiting The detection request is still valid, the keyword has not yet been detected in the unsolicited messages. expired The detection request expired, and the keyword was not found in the unsolicited messages. The detection request expires when the value for detect-time on the issue command request is exceeded. detected The keyword was found in the unsolicited messages.
msg	Returned when the value of status is detected. This is the message that contains the keyword that was detected.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 422. Response content for an unsuccessful get detect result request

Field name	Description
return-code	Category of the error.
reason-code	Specific error.
reason	Text that describes the cause of the error.

Example HTTP interaction

1. The example in [Figure 362 on page 794](#) shows a request to get the results for a detect request. The command was issued with a generated console name. The detection key returned by the issue command request is D002185.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/defcn/detections/D002185
```

Figure 362. Sample request to get the detect result

The following is the response body for the request. The request is still valid, but the keyword has not been found.

```
{"status":"waiting","msg":""}
```

Figure 363. Sample response body for a get detect result request

2. The example in [Figure 362 on page 794](#) shows a request to get the results for a detect request. The command was issued with a generated console name. The detection key that was returned by the issue command request is D122033.

```
GET https://pev076.pok.ibm.com/zosmf/restconsole/consoles/defcn/detections/D122033
```

Figure 364. Sample request to get the detect result

The following is the response body for the request. The keyword was found. In the response, \r is the return character.

```
{"status": "detected", "msg": "BPXM023I (ZOSMFAD)\r CFZ12584W: CIM Runtime Environment Userid currently only has READ\r access to BPX.SERVER. It is recommended to have either UPDATE access\r to BPX.SERVER or has to be UID 0."}
```

Figure 365. Sample response body for a successful get detect result request

Get messages from a hardcopy log

Use this operation to retrieve messages from hardcopy logs on the system.

HTTP method and URI path

```
GET /zosmf/restconsoles/v1/log
```

where:

<version>

The version of the z/OS Console Services. "v1" is the valid value.

[?time=<time>×tamp=<timestamp>&timeRange=<time-range>&direction=<direction>]

Represents the query parameters that are used to qualify the request. For descriptions of the query parameters, see [“Query parameters”](#) on page 795.

Query parameters

Table 423. Query parameters for a Get Messages request		
Parameter	Required or Optional	Description
time	Optional	Specifies when z/OSMF starts to retrieve messages in the ISO 8601 JSON date and time format. For example, 2021-01-26T03:33:18.065Z. The default value is the current UNIX timestamp on the server. This value is used if the timestamp parameter is not specified.
timestamp	Optional	Specifies the UNIX timestamp, which is the number of milliseconds since 1970-01-01 UTC. This parameter is specified, the "time" parameter is ignored.

Table 423. Query parameters for a Get Messages request (continued)

Parameter	Required or Optional	Description
timeRange	Optional	<p>Specifies the time range for which the log is to be retrieved. Supported time units include s, m, and h for seconds, minutes, and hours. For example: 10s, 10m, 10h.</p> <p>The format is nnnu, where nnn is a number 1-999 and u is one of the time units "s", "m", or "h". For example, 999s of 20m.</p> <p>The default is 10m.</p> <p>Note: The maximum return size of the log is 10000. If more than 10000 logs exist in the timeframe, the system returns the first 10000 logs.</p>
hardcopy	Optional	<p>Specify the source where the logs come from.</p> <p>operlog Get log messages from OPERLOG. Case is not significant.</p> <p>syslog Get log messages from SYSLOG. Case is not significant.</p> <p>Note: If the hardcopy parameter is not specified, the API tries OPERLOG first. If the OPERLOG is not enabled on the system, the API returns the SYSLOG to the user.</p>
sysName	Optional	<p>The name of the system on which the SYSLOG resides. This parameter is only valid when the hardcopy is set to SYSLOG. The system specified here must be in the same JES2 SPOOL Data Set with the local system that the z/OSMF server is running on.</p> <p>If the sysname is not specified when the hardcopy=SYSLOG, the API returns the SYSLOG of the current system that z/OSMF server is running on.</p>
direction	Optional	<p>Specifies the direction (from a specified time) in which messages are retrieved. The options are "forward" or "backward." These strings are case-insensitive.</p> <p>The default is "backward," meaning that messages are retrieved backward from the specified time.</p>

Description

This operation retrieves messages from the hardcopy log for a period of time that is specified by optional parameters.

On successful completion, HTTP status code 200 is returned. The response content is described in [Table 425 on page 799](#).

Notes:

1. The maximum return size of a log is 10000. If more than 10000 logs exist in a specified timeframe , the system returns the first 10000 logs.
2. For OPERLOG, z/OS Console service retrieves both active and inactive log data. For more information, see [Browsing both active and inactive data in z/OS MVS Programming: Assembler Services Guide](#).
3. For SYSLOG, z/OS Console service retrieves it from JES2 SPOOL Data Set. For more information, see [JES Spool Data Set Browse in z/OS JES Application Programming](#). Messages in the SYSLOG might not be in chronological order. You must consider this when you retrieve syslog records based on a timestamp. The out-of-order scenario often happens at:

- a. The beginning of the SYSLOG. For example:

```
21:18:11.94 SYSLOG 00000000 IEE042I SYSTEM LOG DATA SET INITIALIZED
21:16:56.48 INTERNAL 00000290 CONTROL M,UEXIT=Y IEAVN701 - INTERNALLY ISSUED K M
21:16:56.51 INTERNAL 00000090 IEA590I WTO USER EXIT IEAVMXIT NOT FOUND
21:16:43.54 00000290 IEA371I SYS1.IPLPARM ON DEVICE 108B SELECTED FOR IPL
PARAMETERS
21:16:43.54 00000290 IEA246I LOAD ID X5 SELECTED
```

- b. The message flood automation. For more information, see [SYSLOG message ordering in z/OS MVS Planning: Operations](#).
4. If the returned log is more than 1 megabyte (1 MB) and the client accepts gzip encoding, the response is compressed.

Request content

None.

Authorization requirements

See [“Required authorizations” on page 764](#).

HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided.

Otherwise, the HTTP status codes in [Table 424 on page 798](#) are returned for the indicated errors.

Table 424. HTTP Error response codes for a Get Messages request

HTTP Status	Return Code	Reason Code	Reason	Descriptions
400	1	22	The parameter s%: s% is invalid. Here are valid examples: s%.	<p>One or more of the following parameters is not valid:</p> <p>time Must be in the ISO-8601 format. Example: 2021-05-25T07:00Z.</p> <p>timeRange The format is nnnU, nnn, in the range 1 - 999, where "U" can be hours "h," minutes "m," or seconds "s". For example: 20h, 543m, 5s.</p> <p>direction Either "backward" or "forward".</p> <p>timestamp Must be a valid UNIX timestamp.</p> <p>Hardcopy Either "operlog" or "syslog".</p> <p>sysName Must be a valid system name in the SYSPLEX.</p>
400	1	23	The time or timestamp specified is a future time. Only a point at the past time is valid.	The specified time or timestamp is in the future. You must specify a time in the past.
400	1	24	The sysName cannot exceed 8 characters.	The value of the sysName field exceeds the maximum length, which is 8 characters. Provide a valid sysName.
500	3	22	TSO/E user region size incorrect, %s	<p>z/OSMF TSO/E service failed to create the TSO/E address space because the region size exceeds the size limit. The context of the error is provided in the message text: %s.</p> <p>To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide them the details of the error.</p>
500	5	6	Cannot retrieve the local time zone.	An error occurred during the retrieval of the local time zone information. For more information, see the log files.
500	8	1	An error occurred during the retrieval of the hardcopy log.	An error occurred during the retrieval of the hardcopy log. For more information, see the log files.

Table 424. HTTP Error response codes for a Get Messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Descriptions
500	12	1	ZlogsException occurred: macro, name=SYSplex.OPERLOG, returnCode=0xn, reasonCode=0xnnn	The macro is IXGCONN or IXGBRWSE. The values in returnCode and reasonCode explain the reason for the exception. For more information, see IXGCONN - Connect/disconnect to log stream in <i>z/OS MVS Programming: Assembler Services Reference IAR-XCT</i> or IXGBRWSE - Browse/read a log stream in <i>z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG</i> .
500	12	2	Failed to retrieve the local system name. Specify the correct sysName in the request.	z/OSMF console service failed to retrieve the local system name. For details about the error, check the z/OSMF logs. Correct any errors that you find. If the problem persists, contact IBM Support and provide them with the details of the error.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the messages and logs. [“Response content” on page 799](#) lists the fields in the JSON object.

Table 425. Response content for a successful Get Messages request

Field name	Description
timezone	Specify the timezone of the z/OS system. Valid values for the timezone range from -12 to 12. For example, "-3" means UTC-3 timezone.
totalItems	Total number of messages returned in the response.
nextTimestamp	The UNIX timestamp. This value might be used in a subsequent request to specify a starting timestamp. Logs in the “nextTimestamp” are not returned in the current response.
items	JSON array of messages. For more information, see Table 426 on page 799 .
source	Indicates the source of the messages. The valid value is "OPERLOG," which indicates the operations log.

Table 426. Messages JSON object

Field name	Description
cart	Eight character command and response token (CART).
color	The color of the message.
jobName	The name of the job that generates the message.
message	The content of the message.

Table 426. Messages JSON object (continued)	
Field name	Description
messageId	The message ID.
replyId	Reply ID, in decimal.
system	Original eight character system name.
type	HARDCOPY.
subtype	Indicate whether the message is a DOM, WTOR, or HOLD message.
time	For example, “Thu Feb 03 03:00 GMT 2021”.
timestamp	UNIX timestamp. For example, 1621920830109.

If a failure occurs, the response body contains a JSON object with a description of the error. [Table 427](#) on [page 800](#) shows the format of the JSON object.

Table 427. Response content for an unsuccessful Get Messages request	
Field name	Description
returnCode	Identifies the category of error.
reasonCode	Identifies the specific error.
reason	Text that describes the cause of the error.

Example HTTP interaction

The example in [Figure 366](#) on [page 800](#) shows a request to retrieve messages from a hardcopy log that occurred during a 1-hour duration.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/v1/log?time=2021-05-25T07:00Z&timeRange=1h
```

Figure 366. Sample request to get messages from a 1-hour duration

The example in [Figure 367](#) on [page 801](#) shows the response body for the preceding request.

```

{
  "nextTimestamp": 1621922666069,
  "source": "OPERLOGS",
  "totalItems": 2,
  "items": [
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163467248",
      "subType": "NULL",
      "time": "Tue May 25 06:04:26 GMT 2021",
      "message": " $HASP395 BPXAS   ENDED - RC=0000",
      "type": "HARDCOPY",
      "cart": "0",
      "timestamp": 1621922666070
    },
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163467760",
      "subType": "NULL",
      "time": "Tue May 25 06:04:26 GMT 2021",
      "message": " $HASP395 BPXAS   ENDED - RC=0000",
      "type": "HARDCOPY",
      "cart": "0",
      "timestamp": 1621922666070
    }
  ]
}

```

Figure 367. Sample response body for a Get Messages request

The example in [Figure 368 on page 801](#) shows a request to retrieve messages that start at a specific timestamp from a hardcopy log.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/v1/log?timestamp=1621920870789&timeRange=15s&direction=forward
```

The example in [Figure 369 on page 801](#) shows the response body for the preceding request.

Figure 368. Sample request for a Get Messages request that starts at a specific timestamp

```

{
  "nextTimestamp": 1621920856259,
  "source": "OPERLOGS",
  "totalItems": 2,
  "items": [
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163454704",
      "subType": "NULL",
      "time": "Tue May 25 05:34:17 GMT 2021",
      "message": " $HASP373 BPXAS   STARTED",
      "type": "HARDCOPY",
      "cart": "0",
      "timestamp": 1621920857500
    },
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163454960",
      "subType": "NULL",
      "time": "Tue May 25 05:34:18 GMT 2021",
      "message": " BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB IZUSVR13 RUNNING IN ASID 0028",
      "type": "HARDCOPY",
      "cart": "0",
      "timestamp": 1621920858120
    }
  ]
}

```

Figure 369. Sample response body for a Get Messages request

z/OS system variable services

The z/OS system variable services are an application programming interface (API), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage z/OSMF variables, retrieve system variables.

Table 428 on page 802 lists the operations that the system variable services provide.

Table 428. z/OS system variable services: operations summary	
Operation name	HTTP method and URI path
“Create or update system variables” on page 803	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
“Get system variables” on page 805	GET /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name> Note: Only “Get system variables” operation supports both the z/OSMF variable and the system symbol. Other operations only support the z/OSMF variable.
“Import system variables” on page 809	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/import
“Export system variables” on page 810	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/export
“Delete system variables” on page 812	DELETE /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>

Table 429 on page 802 describes the variables that can be specified in the system variable services URI paths.

Table 429. z/OSMF system variable services: URI path variables	
URI path variable	Description
<version>	The version of the system variable services API. The following value is valid: 1.0.
<sysplex-name>	The name of the sysplex.
<system-name>	The name of the system.

Authorization requirements

To use the system variable services API, the client must be authenticated. For more information, see [“Authenticating to z/OSMF” on page 3](#).

Also, for some of the z/OSMF system variable services, the client requires READ access to the ZOSMF.VARIABLES.SYSTEM.ADMIN resource profile in the ZMFAPLA class. See the description of the individual APIs for details.

For system symbol services, your user ID must have the authority to issue a command with the z/OS console REST API, for more information, see [“Required authorizations” on page 764](#) for z/OS console services. The console name that the System Symbol Service uses to issue the command is in the format 'IZUSV[0-9A-Z][0-9A-Z][0-9A-Z]', from 'IZUSV000' to 'IZUSVZZZ'.

The required authority is:

- READ access to the MVS.MCSOPER.* resource in the OPERCMDS class.
- READ access to the MVS.DISPLAY.SYMBOLS resource in the OPERCMDS class.
- READ access to the MVS.ROUTE.CMD.<sysname> resource in the OPERCMDS class, where "sysname" is the name of the system on which the system symbol is defined.

Error logging

Errors from the system variable services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For more information, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded.

HTTP 204 No content

The request was processed successfully; however, no content was returned. This status is normal for some types of requests, such as creating or updating system variables.

HTTP 400 Bad request

The request was missing required input, had errors in the provided input, included extraneous input, or cannot be otherwise served. Additional information that regards the error is provided in an error response body that includes a reason code with additional information. Do not repeat the request without first correcting it.

HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 404 Not found

The requested resource does not exist.

HTTP 500 Server error

A server error occurred during processing of the request.

Create or update system variables

Use this operation to create or update z/OSMF system variables in the system variable pool.

HTTP method and URI path

```
POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF system variables service. The following value is valid: 1.0.
- *<sysplex-name>* identifies the sysplex.
- *<system-name>* identifies the system.

Description

This operation creates or updates system variables specified in the request body. If the system variable pool does not exist, this operation creates the pool and adds the variables to it. If there is no request body and the pool does not already exist, the operation creates an empty pool. If there is no request body and the pool already exists, no action is taken. If a variable appears in the request body multiple times, the value of the last occurrence is used as the value of the variable.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the system variables were created or updated with the new value.

Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class:
ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 802.](#)

Request content

The request content is expected to contain an array of JSON objects. See [Table 430 on page 804](#) and [Table 431 on page 804](#). A request with no request body creates an empty pool if the pool does not already exist.

Table 430. Request content for the create or update system variables request		
Field name	Type	Description
system-variable-list	Array of objects	List of variables to be added or updated to the system variable pool.

Table 431. Fields in a JSON object for the create or update system variables request			
Field name	Type	Required or optional	Description
name	String	Required	Descriptive name for the variable
value	String	Required	Value for the variable
description	String	Optional	Description of the variable

HTTP status codes

On successful completion, HTTP status code 204 (no content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 432. HTTP error response codes for a create or update system variables request	
HTTP error status code	Description
400	The request is missing a required property or the value of the property was null or empty, or the JSON request is incorrectly formatted.
401	Submitter of the request is not authorized to invoke the task to create the system variables.
404	Requested <i>sysplex-name.sysname-name</i> was not found.
404	Requested <i>system-name</i> was not found.
500	A server error occurred during processing of the request.

Response content

None.

Example HTTP interaction

In the following example, the POST method is used to create the system variables.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE
```

Figure 370. Sample request to create system variables

The request body is as follows:

```
{
  "system-variable-list": [
    {
      "name" : "var1", "value": "value1", "description": "description of the variable"},
    {
      "name" : "var2", "value": "value2", "description": "description of the variable"}
  ]
}
```

Figure 371. Sample request body for a create system variables request

Get system variables

Use this operation to get the z/OSMF variables or system symbols from a selected system.

HTTP method and URI path

```
GET /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

```
GET/zosmf/variables/rest/<version>/systems/local
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF system variables service. The following value is valid:
1.0.
- **<sysplex-name>** identifies the sysplex. For system symbols, we only support the local sysplex-name here.
- **<system-name>** identifies the system.
- **<local>** indicates the API to retrieve the variable/symbol value from the local system which the z/OSMF server is running on.

Table 433. Request Parameters

Parameter	Required/Optional	Description
var-name	Optional	<p>The name of a z/OSMF variable or a system symbol.</p> <ul style="list-style-type: none"> System symbol name, the name that is assigned to a symbol, a character string. Note that in IEASYMxx, the symbol name begins with an ampersand (&) and optionally ends with a period (.). For this REST API, the ampersand (&) and the period (.) are omitted. System symbols should usually be specified in uppercase. There are places, for example, operator commands, where system symbols can be specified in lowercase. z/OSMF variable name, the name that is assigned to a z/OSMF variable, a character string. <p>If you omit the var-name in the request, all the z/OSMF variable values/symbol values defined in the system are returned.</p> <p>If you need to specify more than one var-name in the request, you can use the array format parameter, for example:</p> <pre>/zosmf/variables/rest/ <version>/systems/<sysplex- name>.<system-name>? var-name =name1&var-name=name2&var- name=name3&var-name=name4&...</pre>
source	Optional	<p>Identify the var-name that represents a z/OSMF variable name or a system symbol name. Valid values for the source are:</p> <ul style="list-style-type: none"> variable Indicates that the var-name is a z/OSMF variable name. This is the default. symbol Indicates that the var-name is a system symbol name.

Description

This operation retrieves z/OSMF variables from the variable pool or system symbols and returns them in a list.

On successful completion, HTTP status code 200 is returned, along with a response body, which is described in [“Response content” on page 807](#).

Request content

None.

Authorization requirements

For general requirements, see [“Authorization requirements” on page 802](#).

Unlike the other z/OSMF Variables services, a get variables request does not require the client to have READ access to ZOSMF.VARIABLES.SYSTEM.ADMIN resource profile in the ZMFAPLA class.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 200 is returned with an empty array of variables.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 434. HTTP error response codes for a get system variables request	
HTTP error status code	Description
404	Requested system was not found.
404	The system identifier in the URI of the request is not valid.
500	A server error occurred during processing of the request.

Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the system variables. See [Table 435 on page 807](#) and [Table 436 on page 807](#). If no system variables match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 435. Get system variables request response body		
Field name	Type	Description
system-variable-list	Array of objects	List of z/OSMF variable value.
system-symbol-list	Array of objects	List of system symbol value.

Note: Either “system-variable-list” or “system-symbol-list” is returned in the response, not both.

Table 436. Get system variables request: objects		
Field name	Type	Description
name	String	Descriptive name for the variable/symbol.
value	String	Value for the variable/symbol.

Table 436. Get system variables request: objects (continued)

Field name	Type	Description
description	String	Description of the variable. For the system-symbol-list, this field is reserved for future use.

Example HTTP interaction

In the following example, the GET method is used to get all of the system variables on a system.

```
GET /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE
```

Figure 372. Sample request to get system variables

```
{system-variable-list: [
  {"name":"sample1", "value":"20", "Description":"value of sample1"},
  {... }
]}
```

Figure 373. Sample response from a get system variables request

In the following example, the GET method is used to get all of the system symbols on a system.

```
GET /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE?source=symbol
```

Figure 374. Sample request to get system symbols

```
{system-symbol-list: [{"name":"sample1", "value":"20", "Description":""}, \{...\ } ]}
```

Figure 375. Sample response from a get system symbols request

In the following example, the GET method is used to get 2 system variables on a system.

```
GET /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE?source=variable&var-name=sample1&var-name=sample2
```

Figure 376. Sample request to get 2 system variables

```
{system-variable-list: [ {"name":"sample1", "value":"20", "Description":"value of sample1"}, {"name":"sample2",
"value":"30", "Description":"value of sample2"}]}
```

Figure 377. Sample response from a get system symbols request

In the following example, the GET method is used to get 2 system symbols on a system.

```
GET /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE?source=symbol&var-name=sample1&var-name=sample2
```

Figure 378. Sample request to get 2 system symbols

```
{system-symbol-list: [ {"name":"sample1", "value":"20", "Description":""}, {"name":"sample2", "value":"30", "Description":""}]}
```

Figure 379. Sample response from a get system symbols request

Import system variables

Use this operation to import z/OSMF system variables from a file.

HTTP method and URI path

POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/import

In this request, the URI path variables are described, as follows:

- <version> identifies the version of the z/OSMF system variables service. The following value is valid: 1.0.
- <sysplex-name> identifies the sysplex.
- <system-name> identifies the system.

Description

This operation imports system variables from a file. The file must be accessible by the authenticated user. The file contains variable definitions in comma-separated value (CSV) format, where each row consists of the variable name, value and description. There should be no header row in the file. The variables imported from the file are processed in the same way as variables that are specified with the create system variables API.

On successful completion, HTTP status code 204 (No content) is returned.

Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class: ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 802](#).

Request content

The request content is expected to contain a JSON object. See [Table 437 on page 809](#).

Table 437. Request content for the import system variables request			
Field name	Type	Required or optional	Description
variables-import-file	String	Required	Path to the CSV-formatted file containing the variables to import

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 438. HTTP error response codes for a create system variables request

HTTP error status code	Description
400	Request body is not syntactically correct
400	Specified file was either not found or could not be opened
400	Specified file has an incorrect format
401	Submitter of the request is not authorized to add or delete system variables
404	Requested system was not found
500	A server error occurred during processing of the request.

Response content

None.

Example HTTP interaction

In the following example, the POST method is used to import system variables from a file.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE/actions/import
```

Figure 380. Sample request to import system variables

The request body is as follows:

```
{ "variables-import-file": "/u/testuser/variables.csv" }
```

Figure 381. Sample request body for an import system variables request

Export system variables

Use this operation to export z/OSMF system variables for a specific system to a file.

HTTP method and URI path

```
POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/export
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF system variables service. The following value is valid:
1.0.
- *<sysplex-name>* identifies the sysplex.
- *<system-name>* identifies the system.

Description

This operation exports system variables, for the system identified in the URI, to a CSV file specified by the request body. It creates the file if it does not exist. Files created by this API can be imported with the import system variables API.

On successful completion, HTTP status code 204 (No content) is returned.

Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class: ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 802.](#)

Request content

The request content is expected to contain a JSON object. See [Table 439 on page 811.](#)

Table 439. Request content for the export system variables request			
Field name	Type	Required or optional	Description
variables-export-file	String	Required	Path to the file to contain the exported system variables. The file must be accessible to the authenticated user.
overwrite	Boolean	Optional	Indicates whether or not the file should be written to if it already exists. If the value is false and the file exists, the call returns with a status code 400. The value defaults to false if it is not specified.

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 440. HTTP error response codes for a create system variables request	
HTTP error status code	Description
400	Request body is not syntactically correct
400	Path is not accessible for writing
400	File exists, but the request did not indicate that it should be written to
404	Requested system was not found
500	A server error occurred during processing of the request

Response content

None.

Example HTTP interaction

In the following example, the POST method is used to export system variables from a file.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE/actions/export
```

Figure 382. Sample request to export system variables

The request body is as follows:

```
{ "variables-export-file": "/u/testuser/backup-variables.csv", "overwrite":true }
```

Figure 383. Sample request body for an export system variables request

Delete system variables

Use this operation to delete z/OSMF system variables from the system variable pool.

HTTP method and URI path

```
DELETE /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF system variables service. The following value is valid:
1.0.
- *<sysplex-name>* identifies the sysplex.
- *<system-name>* identifies the system.

Description

This operation removes system variables from the system variable pool.

If all variables are removed, the system variable pool is empty. If there is no request body, this operation deletes the system variable pool. If the request body contains an empty array ([]), no action is taken. If the request body contains no variables in the array, no action is taken. If the request body contains variables that does not exist in the pool, those variables are ignored.

On successful completion, HTTP status code 204 (No content) is returned.

Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class:
ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 802](#).

Request content

The request content is expected to contain an array of strings. Each string represents the name of a system variable to delete.

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 441. HTTP error response codes for a create system variables request	
HTTP error status code	Description
400	Request body is not formatted correctly
401	Submitter of the request is not authorized to delete the system variables

Table 441. HTTP error response codes for a create system variables request (continued)

HTTP error status code	Description
404	Requested system was not found
500	A server error occurred during processing of the request.

Response content

None.

Example HTTP interaction

In the following example, the DELETE method is used to delete system variables from a system variable pool.

```
DELETE /zosmf/variables/rest/1.0/variables/systems/<sysplex-name>.<system-name>
```

Figure 384. Sample request to delete system variables

The request body is as follows:

```
["var1", "var2", ...]
```

Figure 385. Sample request body for a delete system variables request

z/OS data set and file REST interface

The z/OS data set and file REST interface is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with data sets and UNIX files on a z/OS system.

The z/OS data set and file REST interface services provide a programming interface for working with z/OS data sets and UNIX files. This function is similar to using GET and PUT requests through file transfer protocol (FTP), but secured through traditional z/OS security controls for user authentication and resource authorizations. For setup details, see [“Required authorizations” on page 818](#).

Table 442 on page 813 lists the operations that the z/OS data set and file REST interface services provide.

Table 442. Operations provided through the z/OS data set and file REST interface services.	
Operation	HTTP method and URI path
“List the z/OS data sets on a system” on page 820	GET /zosmf/restfiles/ds? dslevel=<dataset_name_pattern>[&volser=<volser>&start=<dsname>]
“List the members of a z/OS data set” on page 823	GET /zosmf/restfiles/ds/<dataset_name>/member? start=<member>&pattern=<mem-pat>
“Retrieve the contents of a z/OS data set or member” on page 826	GET /zosmf/restfiles/ds/[-(<volser>)]<data-set-name>[(<member-name>)]

Table 442. Operations provided through the z/OS data set and file REST interface services. (continued)

Operation	HTTP method and URI path
“Write data to a z/OS data set or member” on page 831	PUT /zosmf/restfiles/ds/[-(<volser>)]<data-set-name>[(<member-name>)]
“Create a sequential or partitioned data set” on page 837	POST /zosmf/restfiles/ds/<data-set-name>
“Delete a sequential and partitioned data set” on page 840	DELETE /zosmf/restfiles/ds/<data-set-name> DELETE /zosmf/restfiles/ds/-(<volume>)/<data-set-name>
“Delete a partitioned data set member” on page 842	DELETE /zosmf/restfiles/ds/<dataset-name>(<member-name>)
“z/OS data set and member utilities” on page 844	PUT /zosmf/restfiles/ds/<to-data-set-name>
“Access Method Services Interface” on page 848	PUT /zosmf/restfiles/ams
“List the files and directories of a UNIX file path” on page 850	GET /zosmf/restfiles/fs?path=<file-path-name>
“Retrieve the contents of a z/OS UNIX file” on page 855	GET /zosmf/restfiles/fs/<file-path-name>
“Write data to a z/OS UNIX file” on page 859	PUT /zosmf/restfiles/fs/<filepath-name>
“Create a UNIX file or directory” on page 862	POST /zosmf/restfiles/fs/<file-path-name>
“Delete a UNIX file or directory” on page 865	DELETE /zosmf/restfiles/fs/<file-pathname>
“z/OS UNIX file utilities” on page 867	PUT /zosmf/restfiles/fs/<file-path-name>
“List z/OS UNIX Filesystems” on page 874	GET /zosmf/restfiles/mfs/

Table 442. Operations provided through the z/OS data set and file REST interface services. (continued)

Operation	HTTP method and URI path
“Create a z/OS UNIX zFS filesystem” on page 876	POST /zosmf/restfiles/mfs/zfs/<file-system-name>
“Delete z/OS UNIX zFS Filesystem” on page 878	DELETE /zosmf/restfiles/mfs/zfs/<file-system-name>
“Mount a z/OS UNIX file system” on page 879	PUT /zosmf/restfiles/mfs/<file-system-name>
“Unmount a UNIX file system” on page 881	PUT /zosmf/restfiles/mfs/<file-system-name>

Using the Swagger interface

You can use the Swagger interface to display information about the z/OS data sets and files REST APIs. The Swagger interface includes four sections: AMS APIs, Dataset APIs, File APIs, and Filesystem APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Processing overview

The z/OS data set and file REST interface services can be invoked by any client application, running on the local z/OS system or a remote system. Your program (the client) initiates a request to the server through a standard HTTP request method, such as GET or PUT. If the server determines that the request is valid, it performs the requested service and returns an HTTP response to your program.

For a successful request, this response takes the form of an HTTP 2nn status code and, if applicable, a result set that is passed back to your program. Depending on which service is requested, the result set might be returned in a format that requires parsing by your program, such as a JSON object. In other cases, the results might be returned in another format, such as plain text or binary data.

For an unsuccessful request, the server response consists of a non-OK HTTP response code and details of the error, which are provided in the form of a JSON object.

The contents of the JSON objects are described in [“JSON document specifications for z/OS data set and file REST interface requests” on page 883](#).

Note:

If the URL contains a reserved character such as # \$ @, it must be URL-encoded so that it can be escaped. For example:

GET /zosmf/restfiles/ds/SYS1.PROCLIB(#ABC) HTTP/1.1

Should be changed to:

GET /zosmf/restfiles/ds/SYS1.PROCLIB(%23ABC) HTTP/1.1

Common HTTP Request Headers

X-IBM-Async-Threshold = <nnn>

This header can be added to a request to enable support for asynchronous responses with the HTTP status code 202 Accepted. This specifies the number of seconds that the client wants to wait for a response before receiving a 202 Accepted response. This response includes a Location response header with the URL (excluding protocol, host, and port) that can be used with a subsequent GET method request to obtain the results from the original request. Each subsequent request includes its

own X-IBM-Async-Threshold header if additional async responses are required. The value of this header must be an integer 0 - 60 seconds. A value of 0 indicates that an async response is returned if the actual response is not immediately available. If X-IBM-Async-Threshold is specified, then X-IBM-Response-Timeout does not apply and is ignored. A DELETE method request can also be sent to the URL returned in an asynchronous response to abandon the original request and terminate the associated CEA TSO address space.

Example of an Asynchronous request

Request:

```
GET /zosmf/restfiles/ds?dslevel=D10 HTTP/1.1
X-IBM-Async-Threshold: 3
```

Response:

```
202 Accepted
X-Powered-By: Servlet/3.0
Location: /zosmf/restfiles/queue/FS11fae7
X-IBM-Txid: tx000000000000D159
Content-Language: en-US
Content-Length: 0
Date: Fri, 18 Nov 2016 07:17:21 GMT
```

Request:

```
GET GET /zosmf/restfiles/queue/FS11fae7 HTTP/1.1 HTTP/1.1
X-IBM-Async-Threshold: 3
```

Response:

```
200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 39199
X-IBM-Txid: tx000000000000D160
Date: Fri, 18 Nov 2016 07:18:32 GMT
{"items": [
  {"dsname": "D10"},
  {"dsname": "D10.$DATA.SETS"},
  {"dsname": "D10.AAAAA"},
  {"dsname": "D10.AACTIVE.JCL"},
  {"dsname": "D10.AA11797.R1K.D050701A"},
  {"dsname": "D10.AA12484.HDZ11K0"},
  {"dsname": "D10.AA12484.HDZ11K0.TRS0"},
  {"dsname": "D10.ABACKUP.SM02631.FPGA.D14163.T131433"},
  {"dsname": "D10.ABARS1.C.C01V0001"}
]}
```

X-IBM-Response-Timeout = <nnn>

This header specifies the number of seconds that a TsoServlet request runs before a timeout occurs and an exception is returned to the client. This time does not include the time that can be required to start a new CEATSO address space. The default is 30 seconds, and the allowed range for this value is 5 - 600 seconds. An invalid value that is supplied for this header is converted into the closest valid value and the request proceeds.

Note: This timeout does not affect any timeouts that might occur in the z/OSMF WebSphere container or the REST service client.

X-IBM-Request-Acctnum = <string>

This header specifies the account number that is used for the TSO/E logon procedure, which is used for the z/OS data set and file REST interfaces. The valid value is a valid accounting number in your installation.

Note:

- If you do not use this header, the RESTAPI_FILE ACCT value specified in z/OSMF parmlib is used again as a TSO/E logon procedure account number.
- If you use this header and specify a string with a length smaller than 1 or longer than 40, the request will fail. Otherwise, it overrides the RESTAPI_FILE ACCT value specified in z/OSMF parmlib.

This header only takes effect when you create a new TSO. If the request is reusing an existing TSO, this header is ignored.

X-IBM-Request-Proc=<string>

This header specifies the TSO procedure name that is used for the TSO/E logon procedure, which is used for the z/OS data set and file REST interfaces.

Note:

- If you do not use this header, the RESTAPI_FILE PROC value specified in z/OSMF parmlib is used again as a TSO/E logon procedure name.
- If you use this header and specify a string with a length smaller than 1 or longer than 8, the request fails. Otherwise, it overrides the RESTAPI_FILE PROC value specified in z/OSMF parmlib. This header only takes effect when you create a new TSO. If the request is reusing an existing TSO, this header is ignored.

X-IBM-Request-Region=<integer>

This header specifies the TSO region size that is used for the TSO/E logon procedure, which is used for the z/OS data set and file REST interfaces. The region size unit is K.

Note:

- If you do not use this header, the RESTAPI_FILE REGION value specified in z/OSMF parmlib is used again as a TSO/E logon procedure region size.
- If you use this header and specify an integer with a size smaller than 1 or longer than 2096128, the request fails. Otherwise, it overrides the RESTAPI_FILE REGION value specified in z/OSMF parmlib. If you set the value of this header to be less than 65536, the system defaults to TSO region size to 65536K. This header only takes effect when you create a new TSO. If the request is reusing an existing TSO, this header is ignored.

X-IBM-Target-System

This header indicates the target system for the request, where the target system name (nick name) is defined in the Systems table of the local system. The target host must support single-sign-on using an LTPA token. If the target system is the local system, this header is ignored.

Common HTTP Response Headers

X-IBM-Txid = <string>

This header returns the transaction ID that was assigned by z/OSMF to the request. It can be useful for diagnostic purposes to identify z/OSMF log records relating to a failed transaction. The transaction ID is also logged in the TSO address space. The transaction ID should not be used for other purposes; its format can change; and it might not be present in future releases.

Content-Encoding

This header is used to compress the response data. If present, its value indicates which encoding method can be used to decompress the media-type that was specified in the Accept-Encoding header. The supported encoding method is gzip, which is specified as follows:

```
Content-Encoding: gzip
```

The response content is compressed by this method, if the content exceeds 4096 bytes.

Specifying an entity tag with your read and write requests

During request processing, your program's access to the resource (a data set or file) is serialized by z/OS. No other users can read the resource or write to it, thus preventing concurrent updates of the resource from overwriting each other. Serialization, however, ends with the completion of the request. If your program must perform multiple read and write requests on a resource, you require a method of ensuring

that the resource is not modified between your program's requests. Otherwise, you might overlay another user's changes.

To help you to ensure the integrity of a resource between requests, the z/OS data set and file REST interface supports the use of an *entity tag* (or *ETag*) on your requests. Obtained on the initial read (GET) request, the ETag is an identifier that is assigned by the web server to a specific version of the resource. If the resource content changes, a new ETag is associated with the resource.

To determine whether a resource was changed between requests, your program supplies its ETag as a header value on each request. If the ETag matches the current ETag for the resource, the system considers the resource to be unchanged and performs the request. Otherwise, the system fails the request; your program must obtain the ETag again before it can perform the request.

Generally, the process of updating the contents of a z/OS data set or UNIX file is as follows:

1. Retrieve the current content of the resource by using a GET request.
2. The server returns the contents of the resource in the response body as plain text, along with information about the resource, in the response headers:
 - Content-Length header specifies the length of the data that was returned
 - ETag header specifies the ETag that identifies the current version of the resource.
3. Replace the contents of the resource by using a PUT request. The request includes the following headers:
 - A request header to supply the ETag that was returned from the previous GET request on that resource. If the token still matches, the resource was not changed since the previous GET request. If the supplied token does not match a currently valid token, the PUT request fails with an HTTP 412. This response indicates that the host system file was modified in the time since the read operation was performed.
 - Optionally, a request header to specify whether data conversion is required.

For a PUT request, the request body contains the new contents of the file.

After the data is written, the 204 No Content response is returned with a new ETag, for use with any subsequent read or write requests.

Suppose you only want to replace a resource with a new copy, without first reading the contents of the current resource. To overlay the resource, have your program issue the initial GET request to obtain the ETag. Here, you would specify a maximum read amount of 0. Then, have your program issue a PUT request with the ETag and the new data to be written to the resource.

Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. For the detailed format of each JSON object, see ["JSON document specifications for z/OS data set and file REST interface requests" on page 883](#).

Required installation

To enable the z/OS data set and file REST interface services, IBM supplies a default procedure in your z/OSMF order, which you must install before you configure z/OSMF. For information, see [Configure the z/OS data set and file REST services in IBM z/OS Management Facility Configuration Guide](#).

Required authorizations

Generally, your user ID requires the same authorizations for using the z/OS data set and file REST interface services as when you perform these operations through a TSO/E session on the system. For example, listing the members of a z/OS data set through the z/OS data set and file REST interface requires authorization to start TSO on the system and access to the specified data set.

In addition, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <IZU_SAF_PROFILE_PREFIX> in the APPL class.
- READ access to the <IZU_SAF_PROFILE_PREFIX>.*.izuUsers profile in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

Where applicable, further authorization requirements are noted in the descriptions of the individual z/OS data set and file REST interface services.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 893](#).

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 204 No content

Request was processed successfully, however, no content was returned. This status is normal for some types of requests, such as when no data sets or files match the filter criteria, or the specified partitioned data set has no members.

HTTP 206 Partial content

Request was processed successfully, however, only a portion of the available content was received. The request contained the X-IBM-Max-Items header, which limited the amount of content that was returned.

HTTP 304 Not Modified

An ETag token was included in the request. z/OSMF determined that the requested resource did not change since the ETag token was created.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 404 Not found

Requested resource does not exist.

HTTP 405 Method not allowed

Requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

HTTP 412 Precondition failed

The supplied ETag token indicated that the resource was modified since the token was created. Therefore, the request failed the If-Match precondition that was specified in the header.

HTTP 413 Request entity too large

The supplied data is too large to process. Or, the requested resource is too large to return.

HTTP 429 Too many requests

The client submitted too many unsuccessful login attempts.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

HTTP 503 Service unavailable

Server is not available.

Error logging

Errors from the z/OS data set and file REST interface services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

List the z/OS data sets on a system

You can use this operation to list the data sets on a z/OS system. You can filter the returned list of data set names through the specification of high-level qualifiers and wildcards.

HTTP method and URI path

```
GET /zosmf/restfiles/ds/?dslevel=<filter-criteria>[&start=dsname]
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **?dslevel=<dataset-name-pattern>[&volser=<volser>&start=<dsname>]** represents the query parameters used to qualify the request, such as a data set name and, optionally, a volume serial (VOLSER).

Standard headers

None.

Custom headers

Include the following custom HTTP header with this request:

X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

X-IBM-Attributes

This header specifies whether the results are to include the data set base or volume attributes.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

You can specify the following query parameter on this request:

dslevel

The search parameter that identifies the cataloged data sets to be listed. Either the dslevel or volser parameter must be specified and can be a fully qualified data set name or a partial data set name with a filter to display a list of matches. A partial data set name can include:

- One or more high-level qualifiers or name segments
- One or more wildcard symbols: percent sign (%), asterisk (*), or double asterisk (**)
- A percent sign is a single character wildcard.
- An asterisk is any number of characters within a qualifier.
- A double asterisk is any number of characters within any number of lower-level qualifiers.

The parameter values must be URL-encoded, otherwise you may receive an error message. If you use the percent sign (%) as a wildcard to filter the list of data sets returned, you must enter %25 to avoid receiving this error message: URLDecoder: Incomplete trailing escape (%) pattern. For example:

```
GET /zosmf/restfiles/ds?dslevel=sys%25d.*lib HTTP/1.1
```

Notes:

1. The length of the data set name that you specify on the request cannot exceed 44 characters. The length limit includes wildcards, which are treated as one character each. The wildcard %25 is treated as one character.
2. The system appends the following to any filter criteria that you specify: .**
3. Lowercase characters are automatically folded to uppercase.

volser

A parameter that identifies the volume serials to be searched for data sets with names that match the specified **dslevel** parameter. The volume serial is one to six characters. You cannot use wildcard characters for this parameter. If you omit this parameter, the cataloged data set name is returned by default. If this parameter is specified, the data sets on the volume that match the **dslevel** pattern are returned.

start

An optional search parameter that specifies the first data set name to return in the response document. The length of the data set name that you specify cannot exceed 44 characters, and cannot contain wildcards. If the data set name is not found for the given search, then the next data set matching the search will be returned.

X-IBM-Attributes

dsname

Requests that only data set names be returned. If you omit this header, it is set to "dsname".

base

Setting the X-IBM-Attribute to base returns all of the basic attributes for the data set being queried. These attributes are commonly found in the **ISPF List Data set panel**. The base key is mutually exclusive with volser, and dsname.

vol

Setting the X-IBM-Attribute to vol returns the volume where the data set resides. If the dataset is a multi-volume dataset, use the base option to list all the volumes.

,total

The suffix ,total, can be added to request that the "totalRows" property is returned if more data sets than the maximum requested are available.

Required authorizations

See [“Required authorizations”](#) on page 818.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Example request

In the following example, the GET method is used to list all of the cataloged data sets that match the partial name IBMUSER.CONFIG.*.

```
GET /zosmf/restfiles/ds?dslevel=IBMUSER.CONFIG.* HTTP/1.1
```

Example response

A sample response is shown in [Figure 386](#) on page 822.

Response:

```
200 OK
x-powered-by: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 201
Content-Language: en-US
Date: Mon, 23 Nov 2015 09:10:11 GMT
```

Response Body:

```
{
  "items": [
    {
      "dsname": "IBMUSER.CONFIG.FS",
      "dsname": "IBMUSER.CONFIG.FS.DATA",
      "dsname": "IBMUSER.CONFIG.ORIG.FS",
      "dsname": "IBMUSER.CONFIG.ORIG.FS.DATA"
    }
  ],
  "returnedRows": 4,
  "JSONversion": 1
}
```

Figure 386. Example: list all of the data sets.

Example request

The GET method is used to list all of the cataloged data sets with specified base attributes.

```
GET /zosmf/restfiles/ds?dslevel=**&volser=PEVTS2 HTTP/1.1
```

Request Headers:

```
X-IBM-Attributes: 'base'
```

Example response

A sample response is shown in [Figure 387](#) on page 823.

Response:

```
200 OK
x-powered-by: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 714
Content-Language: en-US
Date: Mon, 23 Nov 2015 09:11:46 GMT
```

Response Body:

```
{
  "items": [
    {
      "dsname": "IBMUSER.CONFIG.FS", "catnm": "CATPAK.MASTER.CATALOG", "dsorg": "VS",
      "migr": "NO", "mvol": "N", "vol": "*VSAM*",
      "dsname": "IBMUSER.CONFIG.FS.DATA", "blksz": "?", "catnm": "CATPAK.MASTER.CATALOG",
      "cdate": "2011/08/14", "dev": "3390", "dsorg": "VS", "edate": "***None***", "extx": "1",
      "lrecl": "?", "migr": "NO", "mvol": "N", "ovf": "NO", "rdate": "2015/07/28", "recfm": "?",
      "size": "14250", "spacu": "CYLINDERS", "used": "?", "vol": "CIMSSR",
      "dsname": "IBMUSER.CONFIG.ORIG.FS", "catnm": "CATPAK.MASTER.CATALOG", "dsorg": "VS",
      "migr": "NO", "mvol": "N", "vol": "*VSAM*",
      "dsname": "IBMUSER.CONFIG.ORIG.FS.DATA", "catnm": "CATPAK.MASTER.CATALOG",
      "dev": "3390", "migr": "NO", "mvol": "N", "vol": "PEVTS2",
      "dsname": "IBMUSER.MULTIVOL.DATA", "catnm": "CATPAK.MASTER.CATALOG",
      "dev": "3390", "migr": "NO", "mvol": "Y", "vol": "PEVTS2", "vols": "PEVTS2,PEVTS3,PEVTS4"
    }
  ],
  "returnedRows": 5, "JSONversion": 1
},
  "returnedRows": 4, "JSONversion": 1
}
```

Figure 387. Example: List all of the cataloged data sets with specified base attributes.

List the members of a z/OS data set

You can use this operation to list the members of a z/OS partitioned data set.

HTTP method and URI path

```
GET /zosmf/restfiles/ds/<dataset-name>/member?start=<member>&pattern=<mem-pat>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **/<dataset-name>** identifies the data set for which members are to be listed. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters. You cannot use wildcard characters for this parameter.
- **/member** indicates that member names are to be returned.

Standard headers

None.

Custom headers

Include the following custom HTTP headers with this request:

X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

X-IBM-Attributes

This header is optional.

member

A request that only member names be returned. If you omit this header, it is set to "member".

base

Setting the X-IBM-Attribute to base returns all of the basic attributes for the data set member being queried. These attributes are commonly found in the **ISPF List Data set panel**. The base key is mutually exclusive with member.

,total

The suffix ,total, can be added to request that the "totalRows" property is returned if more data set members than the maximum requested are available.

X-IBM-Migrated-Recall

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

wait

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

nowait

If the data set is migrated, request it to be recalled, but do not wait.

error

If the data set is migrated, do not attempt to recall the data set.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters**start**

An optional search parameter that specifies the first member name to return in the response document. The length of the data set name that you specify cannot exceed 8 characters, and cannot contain wildcards. If the member name is not found for the given search, then the next member matching the search is returned.

pattern

An optional search parameter restricts the returned member names to only the names that match the given pattern. The syntax of this argument is the same as "pattern" parameter of the ISPF LMMLIST service.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For a successful request, the HTTP response includes an array of data set members, each as one of the following types of JSON list document:

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example request

In the following example, the GET method is used to list all of the members of a data set.

```
GET /zosmf/restfiles/ds/SYS1.PROCLIB/member HTTP/1.1
```

Example response

A sample response is shown in [List all of the members of a data set](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 235
Content-Language: en-US
Date: Tue, 24 Nov 2015 05:31:51 GMT
```

Response Body

```
{
  "items": [
    { "member": "CREATECD" },
    { "member": "SPROCLA1" },
    { "member": "TESTJCL" },
    { "member": "WASACR" },
    { "member": "WLMCD" },
    { "member": "XRACFH" },
    { "member": "XRACFHT" },
    { "member": "XRACFH2" }
  ],
  "returnedRows": 8,
  "JSONversion": 1
}
```

Figure 388. Example: List all of the members of a data set

Example request

In the following example, the GET method is used to list all of the members of a data set with specified base attributes.

```
GET /zosmf/restfiles/ds/SYS1.PROCLIB/member HTTP/1.1
```

Request Headers:

```
X-IBM-Attributes: 'base'
```

Example response

A sample response is shown in [List all of the members of a data set with specified base attributes](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 1287
Content-Language: en-US
Date: Tue, 24 Nov 2015 05:33:57 GMT
```

Response Body

```
{
  "items": [
    {
      "member": "CREATECD", "vers": 1, "mod": 0, "c4date": "2015/08/12", "m4date": "2015/08/12", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "05:48", "msec": "43", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "SPROCLA1", "vers": 1, "mod": 12, "c4date": "2009/10/16", "m4date": "2014/09/18", "cnoirc": 132,
      "inoirc": 122, "mnoirc": 0, "mtime": "07:55", "msec": "23", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "TESTJCL", "vers": 1, "mod": 0, "c4date": "2015/07/29", "m4date": "2015/07/29", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "01:49", "msec": "36", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "WASACR", "vers": 1, "mod": 0, "c4date": "2015/08/14", "m4date": "2015/08/14", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "04:44", "msec": "19", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "XRACFH", "vers": 1, "mod": 1, "c4date": "2005/09/26", "m4date": "2005/11/03", "cnoirc": 514,
      "inoirc": 506, "mnoirc": 8, "mtime": "11:10", "msec": "45", "user": "HDENNIS", "sclm": "N"},
    {
      "member": "XRACFHT", "vers": 1, "mod": 0, "c4date": "2005/11/04", "m4date": "2005/11/04", "cnoirc": 130,
      "inoirc": 130, "mnoirc": 0, "mtime": "11:28", "msec": "12", "user": "HDENNIS", "sclm": "N"},
    {
      "member": "XRACFH2", "vers": 1, "mod": 0, "c4date": "2005/11/04", "m4date": "2005/11/04", "cnoirc": 130,
      "inoirc": 130, "mnoirc": 0, "mtime": "11:27", "msec": "43", "user": "HDENNIS", "sclm": "N"}
  ],
  "returnedRows": 8, "JSONversion": 1
}
```

Figure 389. Example: List all of the members of a data set with specified base attributes.

Retrieve the contents of a z/OS data set or member

You can use this operation to retrieve the contents of a sequential data set, or a member of a partitioned data set (PDS or PDSE). To retrieve the contents of an uncataloged data set, include the volume serial on the request.

HTTP method and URI path

```
GET /zosmf/restfiles/ds/[-(<volser>)]<dataset-name>[(<member-name>)]
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **-(<volser>)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified **<data-set-name>** or **<member-name>**. The length of the volume serial cannot exceed six characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.
- **<dataset-name>** identifies the data set to be read. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- **<member-name>** identifies the name of the PDS or PDSE member to be read. Include this parameter for a member read request.

Based on the object to be read, you can specify one of the following parameter combinations:

- **/<data-set-name>**: To retrieve data from a sequential data set.
- **/<data-set-name>(<member-name>)**: To retrieve data from a member of a PDS or PDSE.
- **/-(<volser>)/<data-set-name>**: To retrieve data from an uncataloged sequential data set.
- **/-(<volser>)/<data-set-name>(<member-name>)**: To retrieve data from a member of an uncataloged PDS or PDSE.

Optional Query Parameters

search=<string>

The data set is searched for the first record that contains the string, without respect to case (by default).

Optionally, insensitive=false may be specified for case sensitive matching.

This parameter may not be specified with the research= parameter.

research=<regular-expression>

The data set is searched for the first record that matches the given extended regular expression.

This parameter may not be specified with the search= parameter.

Implementation note: the regcomp() C Library function with the REG_EXTENDED flag is used.

insensitive=true|false

The default is 'true'. When 'true', searches (search and research) are case insensitive. For case sensitive searches, specify 'false'.

maxreturnsize=<integer>

This parameter may be specified only with search= or research=.

The value given is the maximum number of records to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching record. The X-IBM-Record-Range request header may be used to specify the range of records to be searched, but it will not restrict the number of records returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search will begin with the first record. In all cases, an X-IBM-Record-Range=p,q response header will be returned where p is the first matching record and q is the number of records returned.

If no matching records are found, the response header X-IBM-Record-Range=0,0 will be returned.

The parameter may not be used if a request header X-IBM-Data-Type specifies any option except 'text'.

Standard headers

You can include the following standard HTTP header with this request:

If-None-Match

This header is optional; use it to specify the ETag token to be used to correlate this request with a previous request. If the data on the z/OS host has not changed since the ETag token was generated, z/OSMF returns a status of HTTP 304 Not Modified.

For the initial request to the resource, you can omit this header.

Note: If this header is used with very large data sets then performance may be impacted since the data set may have to be read twice by the system. This header is ignored if X-IBM-Record-Range is specified (see below). The ETag response header may be returned containing a hash string. See [“X-IBM-Return-ETag” on page 828](#) for details on whether this header will be present.

Custom headers

You can include the following custom HTTP header with this request:

X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the returned data, as follows:

- When set to text, data conversion is performed. The data transfer process converts each record from EBCDIC to the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. A newline (NL) character from the response charset is inserted between logical records. For data sets with fixed-length records, trailing blanks are removed.

A value "text;fileEncoding=<codepage>" can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

Note: An alternate file encoding cannot be specified with the "research" query parameter.

- When set to `binary`, no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are added between records. The response Content-Type is "application/octet-stream".
- When set to `record`, no data conversion is performed. Each logical record is preceded by the 4-byte big endian record length of the record that follows. This length does not include the prefix length. For example: a zero-length record is 4 bytes of zeros with nothing following.

If you omit this header, the default is `text`; the response is converted.

X-IBM-Return-Etag

This header is optional; set it to 'true' to force the response to include an "Etag" header, regardless of the size of the response data. If this header is not present or set to something other than 'true', then the default is to only send an Etag in the response for data sets smaller than a system determined length, which is at least 8MB. If X-IBM-Record-Range is present, then this header may not be specified with the value "true" and an Etag will never be returned.

If this header is enabled for very large data sets, then performance is impacted since the data set must be read twice by the system.

X-IBM-Migrated-Recall

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

wait

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

nowait

If the data set is migrated, request it to be recalled, but do not wait.

error

If the data set is migrated, do not attempt to recall the data set.

X-IBM-Record-Range

Use this header to retrieve a range of records from a data set. You can specify this range using either of the following formats:

SSS-EEE

Where SSS identifies the start record and EEE identifies the end record to be retrieved. Both values are relative offsets (0-based).

When EEE is set to 0, records through the end of the data set are retrieved.

When SSS is omitted (i.e. -EEE), the final EEE records of the data set are retrieved.

SSS,NNN

Where SSS identifies the start record and NNN identifies the number of records to be retrieved.

NNN must be greater than zero.

Usage notes:

If X-IBM-Record-Range is specified, then an ETag header will not be returned and the If-None-Match request header is ignored.

If X-IBM-Record-Range header is present on the request, then header X-IBM-Return-Etag=true may not be specified.

If no records are found in the range specified, an exception is returned.

X-IBM-Obtain-ENQ

This header is optional; set it to one of the following values to request that a system ENQ be obtained and held after the completion of this request. If not specified, then no ENQs will be held after the completion of this request.

EXCLU

a SYSDSN/Exclusive ENQ will be held on the data set

SHRW

a SYSDSN/SHR ENQ will be held on the data set, and a SPFEDIT/EXCLU ENQ will be held on the data set, including the member name if this is a request for a PDS member.

A successful response will include an X-IBM-Session-Ref response header that can be added as a request header to subsequent requests to specify affinity to the TSO address space holding this ENQ.

X-IBM-Session-Ref

This header is optional; include it with the value returned from a previous X-IBM-Session-Ref response header to indicate that your request should be executed in the TSO address space that was previously reserved with a X-IBM-Obtain-ENQ request header. This address space will not be used for other requests and if not used at least once every 10 minutes it will be terminated.

The following URL request may be used to "ping" the reserved address space to keep it alive:

GET https://zosmf1.yourco.com/zosmf/restfiles/ping HTTP/1.1

X-IBM-Session-Ref: xxxxxx

The X-IBM-Obtain-ENQ and X-IBM-Session-Ref headers are mutually exclusive.

X-IBM-Release-ENQ

This header is optional; it may be specified with a value "true" to request that the ENQ held by the associated TSO address space be released.

This header must be specified along with a valid X-IBM-Session-Ref header.

X-IBM-Dsname-Encoding:

This header is optional. Use it to indicate your data set and member name codepage. One restriction is that the data set or member name character's UTF-8 code can be converted to a valid IBM-1047 character.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 304 indicates unchanged

content when a conditional get is performed (such as when using the **If-None-Match** header with an ETag from a previous response). A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example request

In the following example, the GET method is used to retrieve the contents of the member SMFPRM00 in data set SYS1.PARMLIB.

```
GET /zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00) HTTP/1.1
```

Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 200)
- ETag that you can use on subsequent requests to test for changes to the resource
- Content-Length response header that specifies the amount of data that was returned (in bytes)
- A response body that contains the resource in plain text.

A sample response header is shown in [Figure 390 on page 830](#).

```
200 OK
Etag: B5C6454F783590AA8EC15BD88E29EA63
Content-Type: text/plain; charset=UTF-8
Content-Language: en-US
Content-Length: 1944
Date: Fri, 07 Nov 2014 02:13:07 GMT
Connection: close
```

Figure 390. Example: Returned contents of the SMFPRM00 member of sys1.parmlib

A sample response body is shown in [Figure 391 on page 830](#).

```
ACTIVE                               /*ACTIVE SMF RECORDING*/           00010000
DSNAME(SYS1.&SMFDSN1,SYS1.&SMFDSN2, /*SMF ON 3390 */                     00020000
SYS1.&SMFDSN3)                       /*FT: SYSAQ3, TS: SYSAQ4 */         00030000
NOPROMPT                            /*PROMPT THE OPERATOR FOR OPTIONS*/ 00040000
REC(PERM)                           /*TYPE 17 PERM RECORDS ONLY*/      00050000
MAXDORM(3000)                       /* WRITE AN IDLE BUFFER AFTER 30 MIN*/ 00060000
MEMLIMIT(256M)                      /* 256M FOR 64 BIT APPS */         00061005
STATUS(003000)                     /* WRITE SMF STATS AFTER HALF HOUR*/ 00070000
JWT(0700)                           /* INVOKE EXIT IEFUTL AFTER 7HR 00M*/ 00080002
SID(&SYSNAME),                      /* SYSTEM ID FOR 3084 - SINGLE IMAGE*/ 00090000
LISTDSN                             /* LIST DATA SET STATUS AT IPL*/    00100000
INTVAL(30)                          /* INTVAL OPTION SP430 */          00110000
SYNCVAL(00)                         /* SYNCVAL OPTION SP430 */         00120000
SYS(NOTYPE(19,40,92),               00130001
  EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI,
    IEFUSI,IEFUTL,IEFU29),INTERVAL(010000),DETAIL) 00140000
                                     00150000
                                     00160000
/* WRITE ALL RECORDS AS THE SYSTEM DEFAULT, TAKE ALL KNOWN 00170000
  EXITS, NOTE: JES EXITS CONTROLLED BY JES , THERE IS NO 00180000
  DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32 00190000
  RECORDS AS A DEFAULT FOR TSO */ 00200000
                                     00210000
SUBSYS(STC,NOTYPE(19,40,92),        00220001
  EXITS(IEFU29,IEFU83,IEFU84,IEFUTL), 00230000
  INTERVAL(SMF,SYNC),DETAIL) /*SP430*/ 00240000
                                     00250000
/* WRITE ALL RECORDS AS BY SYSTEM DEFAULT, TAKE ONLY THREE 00260000
  EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A 00270000
  STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. USE ALL OTHER 00280000
  SYS PARAMETERS AS A DEFAULT */ 00290000
```

Figure 391. Example: Returned contents of the SMFPRM00 member of sys1.parmlib

Example request

In the following example, the GET method is used to retrieve the contents of a sequential data set.

```
GET /zosmf/restfiles/ds/JIAHJ.REST.SRVMP HTTP/1.1
```

Example response

A sample response body is shown in [Figure 392 on page 831](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: text/plain; charset=UTF-8
Content-Length: 2131
Etag: 47029CDDCD91E2887E1FAAD6FCD75ECB
Content-Language: en-US
Date: Wed, 25 Nov 2015 02:27:15 GMT
```

Response Body

```
//JH2FPROC EXEC PGM=IKJEFT01,DYNAMNBR=200
//*****
/* TSO LOGON PROC FOR Z/OS DATA SET AND FILE REST INTERFACE */
/*
/* PROPRIETARY STATEMENT:
/*
/* LICENSED MATERIALS - PROPERTY OF IBM
/* 5610-A01
/* COPYRIGHT IBM CORP. 2014
/* STATUS = HSMA210
//*****
//CEEOPPTS DD *
DYNDDUMP(*USERID.PRIVATE)
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC
// DD DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB
// DD DISP=SHR,DSN=SYS1.SBPXEXEC
//ISPLLIB DD DISP=SHR,DSN=JIAHJ.REST.LMOD
//ISPLLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPTLIB DD RECFM=FB,LRECL=80,SPACE=(TRK,(1,0,1))
// DD DISP=SHR,DSN=ISP.SISPTENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPROF DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(15,15,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//IZUSRVMP DD DISP=SHR,DSN=JIAHJ.REST.PARMLIB(IZUSRVMP)
//SYSOUT DD SYSOUT=H
//CEEDUMP DD SYSOUT=H
//SYSUDUMP DD SYSOUT=H
//
```

Figure 392. Example: Retrieve the contents of a sequential data set

Write data to a z/OS data set or member

You can use this operation to write data to an existing sequential data set, or a member of a partitioned data set (PDS or PDSE). To write to an uncataloged data set, include a volume serial on the request.

HTTP method and URI path

```
PUT /zosmf/restfiles/ds/[-(<volser>)]<dataset-name>[(<member-name>)]
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request

- **-(*<volser>*)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified *<data-set-name>* or *<member-name>*. The length of the volume serial cannot exceed six characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.
- ***<dataset-name>*** identifies the data set to which to write. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- ***<member-name>*** identifies the name of the PDS or PDSE member to which to write. Include this parameter for a PDS or PDSE member write request.

If the member does not exist, it is created. If the data set name identifies a base name of a Generation Data Group (GDG), then member may refer to relative data sets, for example: (0), (+1), (-1)

Based on the object to which you want to write, you can specify one of the following parameter combinations:

- ***/<data-set-name>***: To write to a sequential data set.
- ***/<data-set-name>(<member-name>)***: To write to a member of a PDS or PDSE.
- ***/-(<volser>)/<data-set-name>***: To write to an uncataloged sequential data set.
- ***/-(<volser>)/<data-set-name>(<member-name>)***: To write to a member of an uncataloged PDS or PDSE.

Request body

The data to write to the target data set. The data is interpreted according to the content-type as one of binary, text, record or 'diff -e' format according a combination of the "Content-Type" and the value of the X-IBM-Data-Type custom header, if present.

Standard headers

You can include the following standard HTTP header with this request:

If-Match

This header is optional; use it to specify the ETag to be used for correlating this request with a previous request on the same resource. If the resource has not changed since the ETag token was generated, the data is written to the target data set or member. Otherwise, if the resource has been modified, the request is failed with status code HTTP 412.

If you omit this header, the data is always written, regardless of whether the resource is changed.

Custom headers

You can use the following custom HTTP header with this request:

X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the request body.

text

When set to `text`, data conversion is performed. The data transfer process converts each record from the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. Each line of data, delimited by a Line Feed in the request charset, is converted to EBCDIC and written as a record to the data set or member. (The line feed character is removed from the data, and the data is padded with the space character to the end of the record if it is a fixed record size data set. For variable record size data sets, the record is written without padding.) If the record size of the data set is smaller than any line of text, an HTTP 400 is returned with a JSON error document indicating that not all data was written.

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

A value `text;CrLf=true` can be used to control whether each input text line is terminated with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.

A value `text;wrap=true` can be used to support wrapping the data when you write to a F or FB format data set in order to avoid any truncation errors.

Note: When set to 'text' and "Content-Type" is "application/x-ibm-diff-e", the input consists of commands in the same format as produced by the z/OS UNIX 'diff -e' command. These commands are used to add, replace and delete lines in the target data set. The following commands are supported:

```
a
c
d
s/.//
opt : g|<n>, g means global
n means search and replace <n> times
```

Each command may be optionally preceded by a line or line range, as allowed by the z/OS UNIX 'ed' command. If an error is detected while processing a command, status code 500 is returned with an exception.

binary

When set to `binary`, no data conversion is performed. The data is written to the data set without respect to record boundaries. All records will be written at their maximum record length and for fixed length record data sets, the last record will be padded with nulls if required.

record

When set to `record`, no data conversion is performed. Each logical record is preceded by the 4-byte big endian record length of the record that follows. This length does not include the prefix length. For example: a zero-length record would be 4 bytes of zeros with nothing following.

If you omit this header, the default is `text`; the request body is converted.

X-IBM-Migrated-Recall

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

wait

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

nowait

If the data set is migrated, request it to be recalled, but do not wait.

error

If the data set is migrated, do not attempt to recall the data set.

X-IBM-Obtain-ENQ

This header is optional; set it to one of the following values to request that a system ENQ be obtained and held after the completion of this request. If not specified, then no ENQs will be held after the completion of this request.

EXCLU

a SYSDSN/Exclusive ENQ will be held on the data set

SHRW

a SYSDSN/SHR ENQ will be held on the data set, and a SPFEDIT/EXCLU ENQ will be held on the data set, including the member name if this is a request for a PDS member.

A successful response will include an X-IBM-Session-Ref response header that can be added as a request header to subsequent requests to specify affinity to the TSO address space holding this ENQ.

X-IBM-Session-Ref

This header is optional; include it with the value returned from a previous X-IBM-Session-Ref response header to indicate that your request should be executed in the TSO address space that was previously reserved with a X-IBM-Obtain-ENQ request header. This address space will not be used for other requests and if not used at least once every 10 minutes it will be terminated.

The following URL request may be used to "ping" the reserved address space to keep it alive:
GET https://zosmf1.yourco.com/zosmf/restfiles/ping HTTP/1.1
X-IBM-Session-Ref: xxxxxx

The X-IBM-Obtain-ENQ and X-IBM-Session-Ref headers are mutually exclusive.

X-IBM-Release-ENQ

This header is optional; it may be specified with a value "true" to request that the ENQ held by the associated TSO address space be released.

This header must be specified along with a valid X-IBM-Session-Ref header.

X-IBM-Dsname-Encoding:

This header is optional. Use it to indicate your data set and member name codepage. One restriction is that the data set or member name character's UTF-8 code can be converted to a valid IBM-1047 character.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Required authorizations

See [“Required authorizations” on page 818](#).

Request content

Your request must supply the data set content. For an example, see [“Example request” on page 835](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. Status code 201 indicates success if a new PDS member was created. Status code 412 indicates that the document does not match the supplied ETag token on the If-Match header as described above. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See “Error report document” on page 893.

Example request

Suppose that you want to update the contents of the SMFPRM00 member of SYS1.PARMLIB using a PUT request. Figure 393 on page 835 shows an example of the request header that you might use.

```
PUT /zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00)
If-Match: B5C6454F783590AA8EC15BD88E29EA63
Content-Type: text/plain; charset=UTF-8
```

Figure 393. Example: Request header for a write request to the SMFPRM00 member of sys1.parmlib

In Figure 393 on page 835, notice that the optional header If-Match is included. This header is specified with an ETag that was obtained from a previous read request on the parmlib member. Using an ETag in this manner allows for conditional processing; the new member contents are written only when the member has not been modified on the host system since the ETag was generated. If the member was modified, for example, by another user or process, this request is failed with HTTP status code 412.

A sample request body is shown in Figure 394 on page 835. The body contains the new contents of the member.

```
/******  
/* THIS PARMLIB MEMBER CONTAINS CONFIGURATION FOR SMF */  
/******  
ACTIVE /ACTIVE SMF RECORDING*/ 00010000  
DSNAME(SYS1.&SMFDSN1,SYS1.&SMFDSN2, /*SMF ON 3390 */ 00020000  
SYS1.&SMFDSN3) /*FT: SYSAQ3, TS: SYSAQ4 */ 00030000  
NOPROMPT /*PROMPT THE OPERATOR FOR OPTIONS*/ 00040000  
REC(Perm) /*TYPE 17 PERM RECORDS ONLY*/ 00050000  
MAXDORM(3000) /* WRITE AN IDLE BUFFER AFTER 30 MIN*/ 00060000  
MEMLIMIT(256M) /* 256M FOR 64 BIT APPS */ 00061005  
STATUS(003000) /* WRITE SMF STATS AFTER HALF HOUR*/ 00070000  
JWT(0700) /* INVOKE EXIT IEFUTL AFTER 7HR 00M*/ 00080002  
SID(&SYSNAME), /* SYSTEM ID FOR 3084 - SINGLE IMAGE*/ 00090000  
LISTDSN /* LIST DATA SET STATUS AT IPL*/ 00100000  
INTVAL(30) /* INTVAL OPTION SP430 */ 00110000  
SYNCVAL(00) /* SYNCVAL OPTION SP430 */ 00120000  
SYS(NOTYPE(19,40,92), 00130001  
EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI, 00140000  
IEFUSI,IEFUTL,IEFU29),INTERVAL(010000),DETAIL) 00150000  
00160000  
/* WRITE ALL RECORDS AS THE SYSTEM DEFAULT, TAKE ALL KNOWN 00170000  
EXITS, NOTE: JES EXITS CONTROLLED BY JES , THERE IS NO 00180000  
DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32 00190000  
RECORDS AS A DEFAULT FOR TSO */ 00200000  
00210000  
SUBSYS(STC,NOTYPE(19,40,92), 00220001  
EXITS(IEFU29,IEFU83,IEFU84,IEFUTL), 00230000  
INTERVAL(SMF,SYNC),DETAIL) /*SP430*/ 00240000  
00250000  
/* WRITE ALL RECORDS AS BY SYSTEM DEFAULT, TAKE ONLY THREE 00260000  
EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A 00270000  
STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. USE ALL OTHER 00280000  
SYS PARAMETERS AS A DEFAULT */ 00290000
```

Figure 394. Example: Request body for a write request to the SMFPRM00 member of sys1.parmlib

Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 204)
- ETag that you can use on subsequent requests to test for changes to the resource

```
204 No Content
Etag: DE2BE8B8485EB8F1E28D3716DFFE0680
Content-Type: application/json; charset=UTF-8
Content-Language: en-US
Date: Fri, 07 Nov 2014 02:31:39 GMT
```

Example request

The PUT method is used to write the contents of a sequential data set.

```
PUT /zosmf/restfiles/ds/JIAHJ.REST.SRVMP HTTP/1.1
If-Match: B5C6454F783590AA8EC15BD88E29EA42
Content-Type: text/plain; charset=UTF-8
```

Example response

A sample response is shown in [Contents of a sequential data set](#).

Response:

```
204 No Content
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Etag: 39E89731CE27214AE2FE0BB9200DC26
Content-Language: en-US
Date: Wed, 25 Nov 2015 03:10:12 GMT
```

Figure 395. Example: Contents of a sequential data set

Example request

The PUT method is used to write the contents to a member of data set with regular expression.

```
PUT /zosmf/restfiles/ds/SYS1.PROCLIB(JH2FPROC) HTTP/1.1
s/a*b/c*c/g
```

Example response

A sample response is shown in [Figure 396 on page 836](#).

Response:

```
204 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Etag: 8EF84322919807BB003EBF2DE067AD38
Content-Language: en-US
Date: Wed, 11 Oct 2017 05:20:10 GMT
```

Figure 396. Example: Contents of a member of data set with regular expression.

Create a sequential or partitioned data set

You can use this operation to create sequential and partitioned data sets on a z/OS system.

HTTP method and URI path

POST /zosmf/restfiles/ds/<dataset-name>

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set that you are going to create.

Request Body

The request body to create a sequential or partitioned data set is shown in [Request body to create a sequential and partitioned data set](#).

Table 443. Request body to create a sequential or partitioned data set		
Field	Type	Description
volser	String	Volume serial.
unit	String	Device type.
dsorg	String	Data set organization.
alcunit	String	Unit of space allocation.
primary	Integer	Primary space allocation.
secondary	Integer	Secondary space allocation.
dirblk	Integer	Number of directory blocks.
avgbk	Integer	Average block size.
recfm	String	Record format.
blksize	Integer	Block size.
lrecl	Integer	Record length.
storclass	String	Storage class.
mgntclass	String	Management class.
dataclass	String	Data class.
dsntype	String	Data set type.
like	String	Model data set name.

Standard headers

None.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For a successful creating request, 201 Created with no content is returned.

Example request

In the following example, the POST method is used to create a sequential data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS HTTP/1.1
```

Request body:

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PS", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80 }
```

Example response

A sample response is shown in [Example: Create a data set](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 10:54:21 GMT
```

Figure 397. Example: Create a data set

Example request

In the following example, the POST method is used to create a partitioned data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS02 HTTP/1.1
```

Request Body

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PO", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "dirblk": 10, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80 }
```

Example response

A sample response is shown in [Example: Create data set](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 11:14:13 GMT
```

Figure 398. Example: Create data set.

Example request

In the following example, the POST method is used to create a PDSE data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS02 HTTP/1.1
```

Request Body

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PO", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "dirblk": 10, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80, "dsntype": "LIBRARY"
}
```

Example response

A sample response is shown in [Figure 399 on page 840](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 11 Oct 2017 11:14:13 GMT
```

Figure 399. Example: Create PDSE data set.

Delete a sequential and partitioned data set

You can use this operation to delete sequential and partitioned data sets on a z/OS system.

HTTP method and URI path

```
DELETE /zosmf/restfiles/ds/<dataset-name>
```

```
DELETE /zosmf/restfiles/ds/-(<volume>)/<dataset-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set, that you are going to delete.
- **<volume>** is where the data set is resided, when the data set is uncataloged.

Request Body

None.

Standard headers

None.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

X-IBM-Dsname-Encoding

This header is optional. Use it to indicate your data set and member name codepage. One restriction is that the data set or member name character's UTF-8 code can be converted to a valid IBM-1047 character.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

If the request is successfully executed, status code 204 indicates success and no content is returned.

Example request

In the following example, the DELETE method is used to delete a data set.

```
DELETE /zosmf/restfiles/ds/JIAHJ.REST.TEST.DATASET HTTP/1.1
```

Example response

A sample response is shown in [Figure 400 on page 841](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:08:38 GMT
```

Figure 400. Example: Delete a data set

Example request

The DELETE method is used to delete an uncataloged data set.

```
DELETE /zosmf/restfiles/ds/-(ZMF046)/JIAHJ.REST.TEST.DATASET2 HTTP/1.1
```

Example response

A sample response is shown in [Delete uncatalogued data set](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:10:22 GMT
```

Figure 401. Example: Delete uncatalogued data set.

Delete a partitioned data set member

You can use this operation to delete a member of a PDS or PDSE.

HTTP method and URI path

```
DELETE /zosmf/restfiles/ds/<dataset-name>(<member-name>)
```

```
DELETE /zosmf/restfiles/ds/-(volume)/<dataset-name>(<member-name>)
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set that contains a member you are going to delete.
- **<member-name>** is the name of the partitioned data set member, that you are going to delete.
- **<volume>** is where the data set resides, when the data set is uncataloged.

Request Body

None.

Standard headers

None.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

X-IBM-Dsname-Encoding

This header is optional. Use it to indicate your data set and member name codepage. One restriction is that the data set or member name character's UTF-8 code can be converted to a valid IBM-1047 character.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

If the request is successfully executed, status code 204 indicates success and no content is returned.

Example request

In the following example, the DELETE method is used to delete a member of a cataloged partitioned data set.

```
DELETE zosmf/restfiles/ds/JIAHJ.REST.TEST.PDS(MEMBER01) HTTP/1.1
```

Example response

A sample response is shown in [Delete a member of a cataloged partitioned data set](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Tue, 15 Sep 2015 10:36:14 GMT
```

Figure 402. Example: Delete a member of a cataloged partitioned data set

Example request

The DELETE method is used to delete a member of an uncataloged partitioned data set.

```
DELETE zosmf/restfiles/ds/-(ZMF046)/JIAHJ.REST.TEST.PDS.UNCAT(MEMBER01) HTTP/1.1
```

Example response

A sample response is shown in Delete a member of an uncataloged partitioned data set.

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Tue, 15 Sep 2015 11:37:12 GMT
```

Figure 403. Example: Delete a member of an uncataloged partitioned data set

z/OS data set and member utilities

You can use the z/OS data set and member utilities to work with data sets and members. The available operations include: rename data set, rename member, copy data set, copy member, migrate data set, recall a migrated data set, and delete a backup version of a data set.

HTTP method and URI path

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
PUT /zosmf/restfiles/ds/<to-data-set-name>(<member-name>)
```

Figure 404. 'rename' request

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
PUT /zosmf/restfiles/ds/<to-data-set-name>(<member-name>)
PUT /zosmf/restfiles/ds/-(<to-volser>)/<to-data-set-name>
PUT /zosmf/restfiles/ds/-(<to-volser>)/<to-data-set-name>(<member-name>)
```

Figure 405. 'copy' request

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
```

Figure 406. 'hmigrate', 'hrecall', or 'hdelete' request

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface.
- **/ds** indicates a data set request.
- **-(<volser>)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified *<data-set-name>* or *<member-name>*. The length of the volume serial cannot exceed 6 characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.
- **<to-data-set-name>** identifies the target data set name. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- **<member-name>** identifies the target PDS or PDSE member name.

Custom headers

The header Content-Type: application/json; charset={charset-name} must be specified, too.

X-IBM-BPXK-AUTOCVT

This header is optional. Use it to indicate how file auto conversion is handled when using the copy operation to copy text mode data sets to POSIX files. If you omit this header, the system default is taken.

'on' or 'all'

The target file is a candidate for automatic conversion if its TXTFLAG is tagged TEXT and the source data set is type TEXT.

'off'

The target file is not a candidate for automatic conversion.

X-IBM-Migrated-Recall

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values can be specified, too:

wait

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

nowait

If the data set is migrated, request it to be recalled, but do not wait.

error

If the data set is migrated, do not attempt to recall the data set.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Request body

A JSON request document (content-type=application/json, character-encoding=UTF-8) must be supplied in one of the following forms:

Table 444. Request			
Function	Property	Description	Required
hmigrate	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No

Table 444. Request (continued)

Function	Property	Description	Required
hrecall	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No
hdelete	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No
	purge:true false	If true, the function uses the PURGE=YES on ARCHDEL request. If false (default) the function uses the PURGE=NO on ARCHDEL request.	No
rename	request	Indicates the function name.	Yes
	from-dataset	The data set to rename.	Yes
		dsn The source data set name. This is required.	Yes
		member If renaming a member this is the old member name. This is not required.	No
	enq	enq for the "to" data set is only allowed for renaming members. Values can be SHRW or EXCLU. SHRW is the default or PDS members, EXCLU otherwise.	No

Table 444. Request (continued)

Function	Property	Description	Required
copy	request	Indicates the function copy.	Yes
	from-file	The file to copy.	You must choose either from-file or from-dataset.
		filename The absolute source file name. This value is required.	
		type One of "binary executable text". Default is text. This is not required.	
	from-dataset	The data set to copy.	
		dsn The source data set. This is required.	
		member Used to specify a member; "*" means all members. This is not required.	
		volser Can be specified if dsn is not cataloged. This is not required.	
		alias:true false if true, aliases are copied along with main member;if false(default), alias relationships are not maintained. This is not required.	
	enq	Only applicable when from-dataset is specified. With from-file, an error is reported (see note below).This is the enqueue type for the "to" data set. Allowed values are: SHR, SHRW, EXCLU;SHRW is the default for PDS, EXCLU for sequential. The source data set is always enqueued via SHR. Note: When from-file is specified, the target dsn is opened with DISP=OLD (EXCLU) with one exception: If the target is a PDS and the from-file/type is text, the target PDS is enqueued SHRW. This is not required.	No
	replace:true false	Applicable with from-dataset. When from-file specified,ignored unless from-file/type=text. If true, members in the target data set are replaced. If false(default), like named members are not copied and an error is returned.	No

Note: The "to" data set must be a PDS if from-dataset/member is '*' or a <member-name> is specified on the URL. When from-dataset/member is a single member name and the member name is NOT specified on the URL, the 'to' data set is expected to be sequential.

Required authorizations

See [“Required authorizations”](#) on page 818.

Usage considerations

Users of the IEBCOPY data set utility might be accustomed to freeing unused space during a data set copy operation for a partitioned data set (PDS). This function is similar to ISPF option 3.1 compress, in which the unused space occupied by deleted or updated members is removed when a PDS is copied. The z/OS data set and member utilities REST API does not offer an equivalent option for a data set copy operation.

For other usage considerations, see [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example

See [Figure 407 on page 848](#) for an example of renaming a data set.

```
Request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ds/MY.NEW.DSN HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "rename", "from-dataset": {"dsn": "MY.OLD.DSN"}}
```

Figure 407. Example: Rename MY.OLD.DSN to MY.NEW.DSN

Example

See [Figure 408 on page 848](#) for an example of copying a PDS member.

```
Request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ds/MY.NEW.DSN(MYMEM2) HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "copy", "from-dataset": {"dsn": "MY.OLD.DSN", "member": "MYMEM1"}, "replace": true }
```

Figure 408. Example: copy member MYMEM1 from MY.OLD.DSN to MY.NEW.DSN(MYMEM2)

Access Method Services Interface

You can use the Access Method Service (IDCAMS) to provide a REST/JSON interface to IDCAMS. You can use this operation to create a new zFS filesystem.

HTTP method and URI path

```
PUT /zosmf/restfiles/ams
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ams** indicates a request for Access Method Services (IDCAMS) services.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Input Content

Input content in a json document:

Table 445. Input Content		
Property	Description	Required
input	one or more input lines <= 255 in length	Yes
JSONversion:1	JSON Version	No

Note: The size of all input lines plus the number of input lines must be <= 8K.

Response Body

If the request is successfully executed, will return 200 status code (IDCAMS RC<=4). In all cases an application/json document will be returned:

Table 446. Response		
Property	Description	Required
rc	Return code from IDCAMS.	Yes
output	One or more input lines <= 255 in length.	Yes
JSONversion	JSON Version.	No

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 819.

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document”](#) on page 893.

Example

Refer to [Figure 409](#) on page 850 for an example of IDCAMS Access Methods Services.

```
request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ams HTTP/1.1
Content-Type: application/json
Content-Length: nn
{
  "input": [
    "DEFINE CLUSTER(NAME (EXAMPL1.KSDS) VOLUMES(VSER05)) - ",
    "DATA (KILOBYTES (50 5))"],
    "JSONversion":1
  ]
}
```

Figure 409. IDCAMS Access Methods Services

List the files and directories of a UNIX file path

You can use this operation to list the files and directories in a UNIX file path on a z/OS system.

HTTP method and URI path

```
GET /zosmf/restfiles/fs?path=<filepath-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface.
- **/fs** identifies a UNIX file path request.
- **?path=<filepath-name>** is a query parameter that specifies the directory that contains the files and directories to be listed.

Standard headers

None.

Custom headers

You can include the following custom HTTP headers with this request:

X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

X-IBM-Lstat

If the value of this header is "true", a `lstat()` is performed on the path rather than `stat()` and a list containing one item is returned with the `lstat` results. For more information about `lstat()` and `stat()`, see [lstat\(\), lstat64\(\) — Get status of file or symbolic link](#) and [stat\(\), stat64\(\) — Get file information](#) in *z/OS XL C/C++ Runtime Library Reference*.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

Specify the following query parameter on this request:

path

This parameter identifies the UNIX directory that contains the files and directories to be listed. This parameter is required and can consist of one or more directories in the hierarchical file system structure, or a fully qualified file name. A fully qualified file name, which consists of the name of each directory in the path to a file plus the file name itself, can be up to 1023 bytes long. You cannot use wildcard characters for this parameter.

The following list contains sample file path names:

```
/
/bin
/usr/lib/libSM.a
```

Filter parameters: To further qualify your request, you can include one or more *filter* parameters; see Table 447 on page 851. The filter parameters are used if the path parameter refers to a UNIX directory. The filter parameters cause z/OSMF to search the directory tree starting with 'path' and return the results that are filtered by the supplied parameters. If more than one of the filter parameters are specified, a logical AND is performed. That is, all of the supplied filter parameters must be true for an entry to be returned.

Table 447. Filter parameters for a "List UNIX Files and Directories" request	
Parameter	Function
group	Select entries that have a group owner of name. If name is an integer value, select entries that have a group owner of GID.

Table 447. Filter parameters for a "List UNIX Files and Directories" request (continued)

Parameter	Function
mtime	Select entries that were modified with a value of number days ago. If a number is given without a minus sign or plus sign, files that are modified exactly number of days ago are selected. If number is preceded with a plus sign, files modified more than number of days ago are selected. If number is preceded with a minus sign, files modified less than number of days ago are selected.
name	Select entries that match pattern according to the rules of fnmatch(). The supplied pattern is matched against the absolute path of the entry, with behavior similar to the find -name option.
size	Select files that are number long. If number does not include a suffix, the number of the file size in bytes. If number includes a suffix of K, the number size is in 1024-byte blocks; if M, the number size is 1048576-byte blocks, and if G the number size is 1073741824-byte blocks. If number is specified without a plus sign or minus sign, files of the size matching number exactly are selected. If number is preceded with a plus sign, files larger than number are selected. If number is preceded with a minus sign, files smaller than number are selected. If this parameter is specified, only regular files are checked; see the type parameter.
perm	Select entries whose permissions match the value octal_mask. If octal_mask is prefixed by a minus sign, entries that have all of the bits set are present in octal_mask. Otherwise, only select entries whose permission bits match octal_mask exactly. Only the rightmost 12 bits (07777) of octal_mask are used.
type	Select entries based on type, as follows: <ul style="list-style-type: none"> * c Character special file * d Directory * f File * l Symbolic link * p FIFO (named pipe) * s Socket <p>If this parameter is specified with the size parameter, it must be set to 'f'. Sizes that are associated with all other types are unspecified.</p>
user	Select entries that have a user owner of name. If name is an integer value, select entries that have a user owner of UID.

Tree traversal parameters: If you include one or more of the above filter parameters in the request, you can add tree traversal parameters to further control the behavior of the directory tree search. See [Table 448 on page 853](#). If no filter parameters are supplied, any tree traversal parameters are ignored.

Table 448. Tree traversal parameters for a "List UNIX Files and Directories" request

Parameter	Function
depth	<p>The default value for this parameter is 0, which means that all subdirectories under path are listed, regardless of depth. When depth is greater than 1, subdirectories up to the specified depth are listed. When depth is 1, only the files in the path are listed.</p> <p>The name field in the returned JSON document contains the path of the entry, relative to the path query parameter.</p>
limit	<p>The limit parameter specifies the maximum number of items to return.</p> <p>Note: This option is the same as X-IBM-Max-Items, but is added as query parameter for convenience. If both limit and X-IBM-Max-Items are supplied, limit is used.</p>
filesys	<p>The default value for this parameter is same. When set to same, only the sub directories on the same file system as the path parameter are listed. If the value is all, all sub-directories under path are listed.</p>
symlinks	<p>The default value for this parameter is follow. When using the follow value, symbolic links are followed. If the value is set to report, symbolic links are returned, but not followed.</p>

Required authorizations

See ["Required authorizations" on page 818](#).

Usage considerations

See ["Usage considerations for the z/OSMF REST services" on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see ["Error handling" on page 819](#).

For a successful request, the HTTP response includes an array of matching of UNIX files and directories, each as a JSON list document. For the contents, see ["File list document" on page 890](#).

For errors, the HTTP response includes error information as a JSON error report document. See ["Error report document" on page 893](#).

Example request

In the following example, the GET method is used to list the files and directories in the UNIX path /usr:

```
GET /zosmf/restfiles/fs?path=/usr HTTP/1.1
```

Example response

A sample response is shown in [Figure 410 on page 854](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 896
Content-Language: en-US
Date: Tue, 24 Nov 2015 06:12:16 GMT
```

Response Body

```
{
  "items": [
    {
      "name": ".", "mode": "drwxrwxrwx", "size": 8192, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    },
    {
      "name": "..", "mode": "drwxr-xr-x", "size": 8192, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-09-15T02:38:29"
    },
    {
      "name": ".profile", "mode": "-rwxrwxrwx", "size": 849, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-02-13T12:08:29"
    },
    {
      "name": ".sh_history", "mode": "-rw-----", "size": 4662, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-06-06T18:09:28"
    },
    {
      "name": "myFile.txt", "mode": "-rw-r--r-", "size": 20, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    },
    {
      "name": "profile.add", "mode": "-rwxrwxrwx", "size": 888, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-05-07T11:23:08"
    }
  ], "returnedRows": 6, "totalRows": 6, "JSONversion": 1
}
```

Figure 410. Example: Returned list of UNIX files and directories in path /usr

Example request

In the following example, the GET method is used to list a UNIX file.

```
GET /zosmf/restfiles/fs?path=/u/ibmuser/myFile.txt HTTP/1.1
```

Example response

A sample response is shown in [Example: Returned list of UNIX files](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 210
Content-Language: en-US
Date: Tue, 24 Nov 2015 09:16:49 GMT
```

Response Body

```
{
  "items": [
    {
      "name": "/u/ibmuser/myFile.txt", "mode": "-rw-r--r-", "size": 20, "uid": 0, "user": "WSADMIN",
      "gid": 1, "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    }
  ], "totalRows": 1, "returnedRows": 1, "JSONversion": 1
}
```

Figure 411. Example: Returned list of UNIX files

Example response

A sample response is shown in [Figure 412 on page 855](#).

Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 210
Content-Language: en-US
Date: Tue, 10 Oct 2017 09:16:49 GMT
GET /zosmf/restfiles/fs?path=/usr/include&name=f*.h HTTP/1.1
```

Response Body

```
{
  "items": [
    {
      "name": "sys/file.h",
      "mode": "-rw-r--r--",
      "size": 2054,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "metal/float.h",
      "mode": "-rw-r--r--",
      "size": 4954,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "ftpcapi.h",
      "mode": "-rw-r--r--",
      "size": 37,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fcntl.h",
      "mode": "-rw-r--r--",
      "size": 7928,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "features.h",
      "mode": "-rw-r--r--",
      "size": 54149,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fenv.h",
      "mode": "-rw-r--r--",
      "size": 5125,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "float.h",
      "mode": "-rw-r--r--",
      "size": 29480,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fntmsg.h",
      "mode": "-rw-r--r--",
      "size": 7668,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fnmatch.h",
      "mode": "-rw-r--r--",
      "size": 3438,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fp.h",
      "mode": "-rw-r--r--",
      "size": 2293,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "ftw.h",
      "mode": "-rw-r--r--",
      "size": 7906,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    }
  ],
  "returnedRows": 11,
  "totalRows": 11,
  "JSONversion": 1
}
```

Figure 412. Example: List UNIX files with more query parameters.

Retrieve the contents of a z/OS UNIX file

You can use this operation to retrieve the contents of a z/OS UNIX System Services file.

HTTP method and URI path

```
GET /zosmf/restfiles/fs/<filepath-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file request
- **<filepath-name>** identifies the UNIX file to be read. This parameter is required and must consist of a fully qualified path and file name.

Optional Query Parameters

search=<string>

The file is searched for the first line that contains the string, without respect to case (by default).

Optionally, insensitive=false may be specified for case sensitive matching.

This parameter may not be specified with the research= parameter.

research=<regular-expression>

The file is searched for the first line that matches the given extended regular expression.

This parameter may not be specified with the search= parameter.

Implementation note: the regcomp() C Library function with the REG_EXTENDED flag is used.

insensitive=true|false

The default is 'true'. When 'true', searches (search and research) are case insensitive. For case sensitive searches, specify 'false'.

maxreturnsize=<integer>

This parameter may be specified only with search= or research=.

The value given is the maximum number of lines to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching line. The maximum line length supported for text searches is 64K. The X-IBM-Record-Range request header may be used to specify the range of lines to be searched, but it will not restrict the number of lines returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search will begin with the first line. In all cases, an X-IBM-Record-Range=p,q response header will be returned where p is the first matching line and q is the number of lines returned.

If no matching lines are found, the response header X-IBM-Record-Range=0,0 will be returned.

The parameter may not be used if a request header X-IBM-Data-Type specifies any option except 'text'.

Standard headers

You can include the following standard HTTP header with this request:

If-None-Match

This header is optional; use it to specify the ETag token to be used to correlate this request with a previous request. If the data on the z/OS host has not changed since the ETag token was generated, z/OSMF returns a status of HTTP 304 Not Modified.

For an initial request to the resource, you can omit this header.

Range

This header is optional; use this header to retrieve a range of bytes from a file. This header is supported only when X-IBM-Data-Type=binary. Specify this range using the following:
bytes=first-byte-pos "-" last-byte-pos

The first-byte-pos value is the byte-offset of the first byte in a range. The last-byte-pos value is the byte-offset of the last byte in the range. The byte positions specified are inclusive. Byte offsets start at zero. When last-byte-pos is not specified or is zero, the range extends to the end of the file. When the first-byte-pos is not specified, a tail range is returned. Comma separated ranges are not supported. For an initial request to the resource, you can omit this header.

Usage notes: If the range cannot be satisfied, i.e. zero bytes are returned, then a status code of 416 is set.

Examples (assuming a file with 10000 bytes):

bytes=0-499 retrieves the first 500 bytes

bytes=500-999 retrieves the second 500 bytes

bytes=500- retrieves the final 9500 bytes

bytes=-500 retrieves the final 500 bytes

X-IBM-Record-Range

Use this header to retrieve a range of records (lines delimited by '\n') from a file. You can specify this range using either of the following formats:

SSS-EEE

Where SSS identifies the start record and EEE identifies the end record to be retrieved. Both values are relative offsets (0-based). When EEE is set to 0, records through the end of the file are retrieved. When SSS is omitted (i.e. -EEE), the final EEE records of the file are retrieved.

SSS,NNN

Where SSS identifies the start record and NNN identifies the number of records to be retrieved.

Usage notes: If X-IBM-Record-Range is specified with Range an error is reported. If zero bytes are returned due to the range specified, status code 500 is returned.

Custom headers

You can include the following custom HTTP header with this request:

X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the returned data, as follows:

- When set to `text`, data conversion is performed. The data transfer process converts each record from EBCDIC to the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. A newline (NL) character from the response charset is inserted between logical records. For data sets with fixed-length records, trailing blanks are removed.

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

Note: An alternate file encoding cannot be specified with the "research" query parameter.

- When set to `binary`, no data conversion is performed. The data transfer process returns each line of data as-is, without translation.

If you omit this header, the default is `text`; the response is converted.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 304 indicates an unchanged file when a conditional get is performed (such as when using the **If-None-Match** header with an ETag from a previous response). Status code 206 indicates that a part of the file has been returned as a result of a Range header on the request. Accompanying this status code will be a Content-Range header in the form sss-eee/nnnnnn where sss-eee is the byte range that was actually returned and nnnnnn is the length of the file. This status is returned only for the standard range header, not the custom X-IBM-Record-Range header. Status code 416 indicates that zero bytes have been returned due to the Range header on the request. This status is returned only for the standard range header, not the custom X-IBM-Record-Range header. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example request

In the following example, the GET method is used to retrieve the contents of the file inetd.conf in the /etc directory.

```
GET /zosmf/restfiles/fs/etc/inetd.conf HTTP/1.1
```

Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 200)
- ETag that you can use on subsequent requests to test for changes to the resource
- Content-Length response header that specifies the amount of data that was returned (in bytes)
- A response body that contains the resource in plain text.

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: text/plain; charset=UTF-8
Content-Length: 2673
Etag: AEA05EC01C7922ADD5103EBD95FFCC58
Content-Language: en-US
Date: Wed, 25 Nov 2015 03:07:10 GMT
```

A sample response body is shown in [Figure 413 on page 859](#).


```

####                                00000100
# Used to replace /etc/inetd.conf on 2nd level z/OS system          00000101
# so we can telnet/rlogin directly to OMVS using PuTTY consoles.      00000102
####                                00000110
# Internet server configuration database                             00000200
#                                                                    00000300
# (C) COPYRIGHT International Business Machines Corp. 1985, 2001    00000400
# All Rights Reserved                                                00000500
# Licensed Materials - Property of IBM                               00000600
#                                                                    00000700
# US Government Users Restricted Rights - Use, duplication or       00000800
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp. 00000900
#                                                                    00001000
# /etc/inetd.conf                                                    00001100
#                                                                    00001200
#           Internet server configuration database                   00001300
#                                                                    00001400
# $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove     00001500
# unsupported services (FIN APAR OW45915)                            00001600
#                                                                    00001700
# Services can be added and deleted by deleting or inserting a      00001800
# comment character (ie. #) at the beginning of a line              00001900
#                                                                    00002000
#=====00002100
# service | socket | protocol | wait/ | user | server | server program 00002200
# name    | type   |          | nowait|      | program | arguments 00002300
#=====00002400
#                                                                    00002500
#                                                                    00002600
otelnets  stream tcp nowait OMVSKERN /usr/sbin/otelnetsd otelnetsd -l
#sh       stream tcp nowait OMVSKERN /usr/sbin/sshd sshd -i
shell     stream tcp nowait OMVSKERN /usr/sbin/orshd orshd -LV      00002700
login     stream tcp nowait OMVSKERN /usr/sbin/rlogind rlogind -m    00002800
exec      stream tcp nowait OMVSKERN /usr/sbin/orexecd orexecd -LV    00002900

```

Figure 413. Example: Response body for a GET request to the UNIX file `/etc/inetd.conf`

Write data to a z/OS UNIX file

You can use this operation to write data to an existing z/OS UNIX System Services file.

HTTP method and URI path

```
PUT /zosmf/restfiles/fs/<filepath-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file request
- **<filepath-name>** identifies the UNIX file to which to write. This parameter is required and must consist of a fully qualified path and file name. If the file already exists, it can be overwritten. If the file does not exist, it can be created.

Request body

The data to write to the target UNIX file. The data is interpreted according to the content-type as one of binary, text, record or 'diff -e' format according a combination of the "Content-Type" and the value of the X-IBM-Data-Type custom header, if present.

Standard headers

You can include the following standard HTTP header with this request:

If-Match

This header is optional; use it to specify the ETag to be used for correlating this request with a previous request on the same UNIX file. If the file has not changed since the ETag was generated, the request is processed. Otherwise, if the file has been modified, the request is failed with status code HTTP 412.

If you omit this header, the data is always written, regardless of whether the file is changed.

Custom headers

You can include the following custom HTTP header with this request:

X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the data to be written, as follows:

text

When set to `text`, data conversion is performed. The data transfer process converts each byte from the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. Line Feed characters are left intact. This is the default value

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

A value `text;CrLf=true` can be used to control whether each input text line is terminated with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.

Note: When set to 'text' and "Content-Type" is "application/x-ibm-diff-e", the input consists of commands in the same format as produced by the z/OS UNIX 'diff -e' command. These commands are used to add, replace and delete lines in the target data set. The following commands are supported:

```
a
c
d
s/.//
opt : gl<n>, g means global
n means search and replace <n> times
```

Each command may be optionally preceded by a line or line range, as allowed by the z/OS UNIX 'ed' command. If an error is detected while processing a command, status code 500 is returned with an exception.

binary

When set to `binary`, no conversion is performed.

If you omit this header, the default is `text`; the data is converted.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success if an existing file was updated. Status code 201 indicates a success if a new file was created. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example request

In the following example, the PUT method is used to replace the UNIX file `/etc/inetd.conf`.

```
PUT /zosmf/restfiles/fs/etc/inetd.conf HTTP/1.1
If-Match: F4A5A479E78AFD4CFF7DF13937AB82AE
Content-Type: text/plain; charset=UTF-8
```

A sample request body is shown in [Figure 414 on page 862](#).

```

####
# Internet server configuration database
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 2001
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# /etc/inetd.conf
#
#           Internet server configuration database
#
# $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove
# unsupported services (FIN APAR OW45915
#
# Services can be added and deleted by deleting or inserting a
# comment character (ie. #) at the beginning of a line
#
#=====
# service | socket | protocol | wait/ | user | server | server program
# name    | type   |          | nowait|      | program | arguments
#=====
#
# Following line uncommented by USSSETUP job: 2013/04/24 15:04:00
otelnet  stream tcp nowait IBMUSER  /usr/sbin/otelnetd otelnetd -l
# Following line uncommented by USSSETUP job: 2013/04/24 15:04:00
shell    stream tcp nowait IBMUSER  /usr/sbin/orshd orshd -LV
# Following line updated by USSSETUP job: 2013/04/24 15:04:00
login    stream tcp nowait IBMUSER  /usr/sbin/rlogind rlogind -m
# Following line added by USSSETUP job: 2013/04/24 15:04:00
ssh      stream tcp nowait IBMUSER  /usr/sbin/sshd sshd -i
#exec    stream tcp nowait OMVSKERN /usr/sbin/orexecd orexecd -LV
# All users should use this configuration file

```

Figure 414. Example: Request body for a PUT request to the UNIX file /etc/inetd.conf

Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 204)
- ETag that you can use on subsequent requests to test for changes to the UNIX file

```

204 No Content
Etag: S8WNSD09SNSNE09B
Content-Type: application/json; charset=UTF-8
Content-Language: en-US
Date: Fri, 07 Nov 2014 02:31:39 GMT

```

Create a UNIX file or directory

You can use this operation to create a UNIX file or directory.

HTTP method and URI path

```
POST /zosmf/restfiles/fs/<file-path>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file or directory.

- **<file-path>** is the name of the file or directory you are going to create.

Request Body

The request body to create a UNIX file or directory is shown in [Request body to create a UNIX file or directory](#).

Table 449. Request body to create a UNIX file or directory		
Field	Type	Description
type	String	The request type. This field supports the values: directory or dir to create a directory. The value: file is supported to create a file.
mode	String	Specifies the file or directory permission bits to be used in creating the file or directory. The characters used to describe permissions are: r: Permission to read the file w: Permission to write on the file x: Permission to execute the file -: No permission An example would be: rwxrwxrwx The nine characters are in three groups of three; they describe the permissions on the file or directory. The first group of 3 describes owner permissions; the second describes group permissions; the third describes other (or world) permissions.

Standard headers

None.

Custom headers

You can include the following custom HTTP headers with this request:

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For a successful create request, 201 Created is returned.

Example request

In the following example, the POST method is used to create a UNIX file.

```
POST /zosmf/restfiles/fs/u/jiahj/text.txt HTTP/1.1
```

Request body:

```
{"type": "file", "mode": "RWXRW-RW-"}
```

Example response

A sample response is shown in [Create a UNIX file](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 30 Sep 2015 11:46:21 GMT
```

Figure 415. Example: Create a UNIX file

Example request

The POST method is used to create a UNIX directory.

```
POST /zosmf/restfiles/fs/u/jiahj/testDir HTTP/1.1
```

Request body

```
{"type":"directory","mode":"rwxr-xrwx"}
```

Example response

A sample response is shown in [Create a UNIX directory](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 30 Sep 2015 10:50:21 GMT
```

Figure 416. Example: Create a UNIX directory

Delete a UNIX file or directory

You can use this operation to delete a UNIX file or directory.

HTTP method and URI path

```
DELETE /zosmf/restfiles/fs/<file-pathname>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file or directory.
- **<file-pathname>** is the name of the file or directory you are going to delete.

Request Body

None.

Standard headers

None.

Custom headers

X-IBM-Option: An optional parameter for deleting a directory. If it is not specified, only the empty directory can be deleted. If it is specified as recursive, it means all the files and sub-directories will be deleted.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

If the request is successfully executed, status code 204 indicates success and no content returned is returned.

Example request

In the following example, the DELETE method is used to delete a UNIX file.

```
DELETE /zosmf/restfiles/fs/u/jiahj/text.txt HTTP/1.1
```

Example response

A sample response is shown in [Figure 417 on page 866](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:10:22 GMT
```

Figure 417. Example: Delete a UNIX file

Example request

The DELETE method is used to delete a UNIX directory.

```
DELETE /zosmf/restfiles/fs/u/jiahj/testDir HTTP/1.1
```

Example response

A sample response is shown in [Delete a UNIX directory](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:15:22 GMT
```

Figure 418. Example: Delete a UNIX directory

z/OS UNIX file utilities

You can use the z/OS UNIX file utilities to operate on a UNIX System Services file or directory. Operations include: chmod, chown, chtag, copy, extattr, getfacl, move, and setfacl.

HTTP method and URI path

```
PUT /zosmf/restfiles/fs/<file-path-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX System Services file system request
- **<file-path-name>** identifies the UNIX file or directory to be the target of the operation. This is required and must consist of a fully qualified path and file or directory name.

Custom headers

X-IBM-BPXK-AUTOCVT

This header is optional. Use it to indicate how file auto conversion is handled when using the copy operation to copy text mode data sets to POSIX files, if you omit this header, the system default is taken.

'on' or 'all'

The target file is a candidate for automatic conversion if its TXTFLAG is tagged TEXT and the source data set is type TEXT.

'off'

The target file is not a candidate for automatic conversion

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-

Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance”](#) on page 432 was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

The header, Content-Type: application/json; charset={charset-name}, must be specified as well.

Request body

A JSON request document (content-type=application/json, character-encoding=UTF-8) must be supplied in one of the following forms:

<i>Table 450. JSON request document</i>			
Function	Property	Description	Required
chmod	request	Indicates the function chmod.	Yes
	mode	The mode value, which is specified as the POSIX symbolic form or octal value (as a JSON string).	Yes
	links:"follow suppress"	The default is 'follow' encountered links. This applies a mode change to the file or directory pointed to by any encountered links. 'suppress' is a mode change for the file or directory pointed to by any encountered symbolic links.	No
	recursive:true false	The default is false. When 'true', the file mode bits of the directory and all files in the file hierarchy below it are changed (chmod -R).	No
chown	request	Indicates the function chown.	Yes
	owner	The user ID or UID (as a JSON string).	Yes
	group	The group ID or GID (as a JSON string).	No
	links:"follow change"	The default is 'follow'. 'change' does not follow the link, but instead changes the link itself (chown -h).	No
	recursive:true false	The default is false. When 'true', changes all the files and subdirectories in that directory to belong to the specified owner (and group, if :group is specified).chown -R)	No

Table 450. JSON request document (continued)

Function	Property	Description	Required
chtag	request	Indicates function chtag.	Yes
	action:"set remove list"	The file tag action. If "set", the file is tagged as specified in the "type" keyword. If "remove", any existing file tag is removed. If "list", the existing tag information will be returned in a JSON response document. See "list action" .	Yes
	type:"binary mixed text"	The default is "mixed" This option can be specified only when the action is "set".	No
	codeset	Specifies the coded character set in which text data is encoded, such as ASCII or EBCDIC. For example, the code set for ASCII is ISO8859-1; the code set for EBCDIC is IBM-1047.	No
	links:"change suppress"	The default is 'change' encountered links, applying a tag action to the file or directory pointed to by any encountered links. 'suppress' a tag action for the file or directory pointed to by any encountered	No
	recursive:true false	The default is false. When 'true', tags all the files and subdirectories in that directory (chtag -R).	No

Note: If the 'list' action is specified, a response JSON document is returned listing the current tag information, For example:

```
{ "stdout": [ "m ISO8859-1  T=off /tmp/file" ] }
```

The -q and -v options are not supported.

Table 450. JSON request document (continued)

Function	Property	Description	Required
copy	request	Indicates the function copy.	Yes
	from:file-or-directory	The file or directory to copy.	You can use either from:file-or-directory or from-dataset.
	from-dataset	dsn The fully qualified data set name. This is required.	
		member The data set member to copy. This is not required.	
		type One of "binary executable text". If not specified, the format of the data set is checked to try to determine the type. This is not required.	
	overwrite:true false	The default is true. May not be specified with 'from-dataset'.	No
	recursive:true false	The default is false. When 'true', copies all the files and subdirectories that are specified by source into a directory (cp -R). May not be specified with 'from-dataset'.	No
	links:"none src all"	The default is none. When 'src', follows symbolic links that are specified as source file or directory (cp -H). When 'all', follows symbolic links specified as source file/directory and those encountered in the tree traverse (cp -L). Cannot be specified with 'from-dataset'.	No
	preserve: "none modtime all"	The default is none, sets the modification time of the destination file to the present. When 'modtime', sets the modification and access time of each destination file to that of the corresponding source file. (cp -m). When 'all', preserves the modification and access times as well as the file mode, file format, owner, and group owner (cp -p). May not be specified with 'from-dataset'.	No
Note: When from-dataset/type == text, and the header X-IBM-BPXK-AUTOCVT == ON ALL, the cp "-O u" switch is supplied to allow automatic conversion. If the from-dataset/type attribute is not specified, no "-O u" switch is applied and automatic conversion will not be available.			
extattr	request	Indicate the function extattr.	Yes
	set:"attrs"	One or more of the following: alps.	No
	reset:"attrs"	One or more of the following: alps.	No
Note: If neither set or reset are provided, a response JSON document is returned listing the attributes, For example:			
<pre> { "stdout": ["/etc/inetd.conf", "APF authorized = NO", "Program controlled = NO", "Shared address space = YES", "Shared library=NO"] } </pre>			
The -F option is not supported.			

Table 450. JSON request document (continued)

Function	Property	Description	Required
getfacl	request	Indicates the function getfacl.	Yes
	type:"access dir file"	The default is 'access', displays the access ACL entries for a file or directory (getfacl -a). 'dir' displays the directory default ACL entries (getfacl -d). If the target is not a directory, a warning is issued.	No
	user	The user ID or UID (as a JSON string), displays only the ACL entries for the specified types of access control lists (getfacl -a, -d, -f) which affects the specified user's access (getfacl -e user).	No
	use-commas:true false	The default is 'false'. When true, displays each ACL entry, using commas to separate the ACL entries instead of newlines.	No
	suppress-header:true false	The default is 'false'. When true, the comment header (the first three lines of each file's output) is not to be displayed (getfacl -m).	No
	suppress-baseacl:true false	The default is 'false'. When true, displays only the extended ACL entries. Does not display the base ACL entries (getfacl -o).	No
Note: On completion of this request, a response JSON document is returned, For example: <pre>{ "stdout": ["#file: /etc/inetd.conf", "#owner: CFZSRV", "#group: SYS1", "user::rwx", "group::rwx", "other::rwx"] }</pre>			
move	request	Indicates the function move.	Yes
	from	The file or directory to be moved.	Yes
	overwrite:true false	The default is true. May not be specified with 'from-dataset'.	No

Table 450. JSON request document (continued)

Function	Property	Description	Required
setfacl	request	Indicates the function setfacl.	Yes
	abort:true false	The default is false. When true, aborts processing if an error or warning occurs. See the setfacl command documentation for complete documentation on the errors and warnings (setfacl -a).	No
	links:"follow suppress"	The default is 'follow'. 'suppress' does not follow symbolic links. Because ACLs are not associated with symbolic links, nothing happens if a symbolic link is encountered (setfacl -h). Note: At least one of the following four keywords must be specified. 'modify' and 'delete' may both be specified, but not with 'delete-type' and 'set'.	No
	delete-type	Delete all extended ACL entries by type (setfacl -D type): access Access ACL dir Directory default ACL file File default ACL every Every extended ACL for all ACL types that are applicable for the current path name. Note: The 'delete-type' keyword cannot be specified with 'set', 'modify' or 'delete'.	No
	set	sets (replaces) all ACLs with 'entries'. 'entries' represents a string of ACL entries. Refer to the setfacl command reference for the string format (setfacl -s entries). Note: The 'set' keyword cannot be specified with 'delete-type', 'modify' or 'delete'.	No
	modify	Modifies the ACL entries. 'entries' represents a string of ACL entries. Refer to the setfacl command reference for the string format. If an ACL entry does not exist for a user or group that is specified in 'entries', it is created. If an ACL entry exists for a user or group that was specified in 'entries', it is replaced. Note: The 'modify' keyword cannot be specified with 'delete-type' or 'set'.	No
	delete	Deletes the extended ACL entries that are specified by 'entries'. 'entries' is a string of ACL entries. Refer to the setfacl command reference for the string format. If an ACL entry does not exist for the user or group specified, no error is issued. Note: The 'delete' keyword cannot be specified with 'delete-type' or 'set'.	No

Table 450. JSON request document (continued)

Function	Property	Description	Required
link	request	Indicates the function link.	Yes
	from	The file/directory to link.	Yes
	type	Indicates the link type as a symbol link or an external link.	Yes
	recursive:true false	The default is false. When "true", it links the files recursively, linking all the files and subdirectories specified by the source into a directory (ln -R).	No
	force:true false	The default is false. When it is "true", it forces a link between files and deletes any conflicting path names that do not have confirmation (ln -f).	No
unlink	request	Indicates the function unlink.	Yes

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 893](#).

Example

Refer to [Figure 419 on page 873](#) for an example of renaming a UNIX file.

```
Example request for renaming /etc/inetd.conf to /etc/inetd.conf.bak:
PUT https://zosmf1.yourco.com/zosmf/restfiles/fs/etc/inetd.conf.bak HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "move", "from": "/etc/inetd.conf"}
```

Figure 419. Example: Rename a UNIX file

List z/OS UNIX Filesystems

You can use the list z/OS UNIX filesystems operation to list all mounted filesystems, or the specific filesystem mounted at a given path, or the filesystem with a given Filesystem name.

HTTP method and URI path

```
GET /zosmf/restfiles/mfs/  
GET /zosmf/restfiles/mfs/?path=file-path-name  
GET /zosmf/restfiles/mfs/?fsname=file-system-name
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs** indicates a Unix System Services filesystem(s) request.

A trailing "/" may optionally be specified.

Query parameters

path

This parameter identifies the UNIX directory that contains the files and directories to be listed. This parameter may not be specified if the 'fsname' parameter is specified. It can consist a directory or fully qualified path name in the UNIX file system structure. A fully qualified file name can be up to 1023 bytes long. You cannot use wildcard characters for this parameter.

fsname

This parameter identifies the fully qualified filesystem name to be listed. For zFS filesystems, this is the data set name of the aggregate. This parameter may not be specified if the 'path' parameter is specified.

Custom headers

X-IBM-MAX-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Required authorizations

See [“Required authorizations”](#) on page 818.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. Status code 404 indicates that the specified filesystem was not found. A JSON response document with no filesystem items will be returned. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 819.

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document”](#) on page 893.

Example

Refer to [Figure 420](#) on page 875 for an example of renaming a data set and PDS member.

```
JSON response document:
Request:
GET https://zosmf1.yourco.com/zosmf/restfiles/mfs/?path=/usr/local HTTP/1.1
GET https://zosmf1.yourco.com/zosmf/restfiles/mfs/?fsname=OMVS.USR.LOCAL.ZFS HTTP/1.1

JSON response document:
{
  "items": [
    {
      "name": "OMVS.USR.LOCAL.ZFS",
      "mountpoint": "/usr/local",
      "fsname": "ZFS",
      "status": "active",
      "mode": ["acl", "synchronly"],
      "dev": 52,
      "fstype": 1,
      "bsize": 1024,
      "bavail": 5615,
      "blocks": 9600,
      "sysname": "SYSNAME1",
      "readibc": 2,
      "writeibc": 0,
      "diribc": 0
    }
  ],
  "JSONversion": 1
}

For a non-specific request to list all filesystems, the following top-level attributes are also returned:
returnedRows - the number of filesystem items returned
totalRows - the total number of filesystems
```

Figure 420. List UNIX Filesystems

If more items than specified by X-IBM-Max-Items (or the default of 1000) match the request, then the following top-level attribute will be added to the top-level document:

```
"moreRows": true
```

JSON response attributes:

These attributes are the mnt3_* values returned from the w_getmntent() C-Library API.

Create a z/OS UNIX zFS filesystem

You can use this operation to create a z/OS UNIX zFS Filesystem.

HTTP method and URI path

```
POST /zosmf/restfiles/mfs/zfs/<file-system-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs/zfs** a UNIX System Services filesystem request for a zFS aggregate. *<file-system-name>* is the filesystem (for zFS, the aggregate name) of the file system to be created. This is also the VSAM linear data set name.

Optional Query Parameters

timeout={secs}

The number of seconds to wait for the "zfsadm format" command to complete before timing out with Category/RC/REAS = 1/8/9 "Command timed out". The default if not specified is 20 seconds. If a greater value is used, the "X-IBM-Async-Threshold" header should be used to allow the application process long running transactions.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Input Content

Input content is a JSON document:

Table 451. Input Content	
Property	Description
owner	Defaults to 755. This property is not required.

Table 451. Input Content (continued)	
Property	Description
group	Defaults to 755. This property is not required.
perms	Defaults to 755. This property is not required.
cylsPri	Defaults to 0. This property is required.
cylsSec	Defaults to 0. This property is not required.
storageClass	This property is not required.
managementClass	This property is not required.
dataClass	This property is not required.
volumes	This property is not required.
JSONversion:1	This property is not required.

Required authorizations

See [“Required authorizations”](#) on page 818.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 819.

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document”](#) on page 893.

Example

Refer to [Figure 421](#) on page 877 for an example of creating UNIX file systems.

```
request:
POST https://zosmf1.yourco.com/zosmf/restfiles/mfs/zfs/HLQ.MYNEW.ZFS HTTP/1.1
Content-Type: application/json
Content-Length: 86
{
  "cylsPri":100,
  "cylsSec": 10,
  "volumes": [ "ZFS001", "ZFS002"],
  "JSONversion":1
}

response:
201 Created
```

Figure 421. Create UNIX Filesystems

Delete z/OS UNIX zFS Filesystem

You can use the delete z/OS UNIX zFS Filesystem operation to delete an existing zFS filesystem. Access Method Services are used to delete the filesystem linear data set. The file system must not be allocated (attached or mounted) for this operation to succeed.

HTTP method and URI path

```
DELETE /zosmf/restfiles/mfs/zfs/<file-system-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs/zfs** a UNIX System Services filesystem request for a zFS aggregate. *<file-system-name>* is the filesystem (for zFS, the aggregate name) of the file system to be deleted. This is also the VSAM linear data set name.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document”](#) on page 893.

Example

Refer to [Figure 422 on page 879](#) for an example of deleting UNIX Filesystems.

```
request:
DELETE https://zosmf1.yourco.com/zosmf/restfiles/mfs/zfs/HLQ.MYNEW.ZFS HTTP/1.1

response:
204 No Content
```

Figure 422. Delete UNIX Filesystems

Mount a z/OS UNIX file system

You can use this operation to mount a z/OS UNIX file system on a specified directory.

HTTP method and URI path

```
PUT /zosmf/restfiles/mfs/<file-system-name>
```

Where:

- /zosmf/restfiles specifies the z/OS data sets and files REST interface.
- /mfs is used for managing file systems.
- <file-system-name> is the file system that you want to mount.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations”](#) on page 818.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Query parameters

None.

Standard headers

None.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-

System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together. Otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request.
- The service that is described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Request Body

The request body to mount a UNIX file system is shown in [Table 452 on page 880](#).

<i>Table 452. Request body to mount a UNIX file system</i>		
Field	Type	Description
action	String	Specifies the action mount to mount an UNIX file system.
mount-point	String	Specifies the mount point to be used for mounting the UNIX file system.
fs-type	String	Specifies the type of file system to be mounted. This value must match the TYPE operand on a FILESYSTYPE statement in the BPXPRMxx parmlib member for your system.
mode	String	<p>Specifies the mode in which the file system is mounted, as follows:</p> <ul style="list-style-type: none"> • Specify <code>rdonly</code> for read only. • Specify <code>rdwr</code> for read/write. • Specify <code>nosetuid</code> or <code>nosetuid rdonly</code> or <code>rdonly nosetuid</code> for read only <code>nosetuid</code>. • Specify <code>nosetuid rdwr</code> or <code>rdwr nosetuid</code> for read write <code>nosetuid</code>. <p>The values are case-insensitive.</p> <p>If not specified, this value defaults to <code>rdonly</code>.</p>

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. Status code 204 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For details, see [“Error handling” on page 819](#).

If the request is successfully run, status code 204 indicates success and no content is returned.

Example request

In the following example, the PUT method is used to mount a UNIX file system.

```
PUT /zosmf/restfiles/mfs/JIAHJ.ZOSMF.DRIVER.HFS HTTP/1.1
Content-Type: application/json; charset=UTF-8
```

Request body

```
{"action": "mount", "mount-point": "/u/jiahj", "fs-type": "HFS", "mode": "rdonly"}
```

Example response

A sample response is shown in [Figure 423 on page 881](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Thu, 17 Sep 2015 08:05:41 GMT
```

Figure 423. Example: Mount a UNIX file system

Unmount a UNIX file system

You can use this operation to unmount a UNIX file system on a specified directory.

HTTP method and URI path

```
PUT /zosmf/restfiles/mfs/<file-system-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **<file-system-name>** is the file system you want to unmount.

Request Body

```
{
  "action" : "unmount"
}
```

Request body to unmount a UNIX file system

The request body to unmount a UNIX file system is shown in [Request body to unmount a UNIX file system](#).

Table 453. Request body to unmount a UNIX file system

Field	Type	Description
action	String	Unmount.

Standard headers

None.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Content type

The content type is application/json.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 819](#).

If the request is successfully executed, status code 204 indicates success and no content returned is returned.

Example request

In the following example, the PUT method is used to unmount a UNIX file system.

```
PUT /zosmf/restfiles/mfs/JIAHJ.ZOSMF.DRIVER.HFS HTTP/1.1
Content-Type: application/json; charset=UTF-8
```

Request body

```
{"action": "unmount"}
```

Example response

A sample response is shown in [Unmount a UNIX file system](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Thu, 17 Sep 2015 08:09:21 GMT
```

Figure 424. Example: Unmount a UNIX file system

JSON document specifications for z/OS data set and file REST interface requests

This section describes the contents of the JSON documents that are used with z/OS data set and file REST interface requests.

For more information about the properties described in these JSON documents, see *z/OS ISPF Services Guide*.

The following JSON documents are described:

- [“Data set list document” on page 883](#)
- [Data set list with attributes document](#)
- [PDS/PDSE member list with attributes document](#)
- [Create a sequential and partitioned data set document](#)
- [“File list document” on page 890](#)
- [Unix file and directory list with attributes document](#)
- [“Mount and unmount a file system document” on page 891](#)
- [Create a UNIX file document](#)
- [“Error report document” on page 893.](#)

Data set list document

[Table 454 on page 884](#) shows the contents of the JSON data set list document.

For more information about these properties, see the dialog variables for the YES | NO | PRT parameter of **LMDLIST** (list data sets) in [LMDLIST-list data sets](#) in *z/OS ISPF Services Guide*.

Table 454. Contents of the JSON data set list document

Property	Description	Required
items	An array where each element contains the following key:value pairs	Yes
dsname	Data set name.	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes
vol	Volume serial	No
used	Percentage of used tracks or pages (PDSE)	No
extx	Number of extents used, long format (5 bytes)	No
cdate	Creation date	No
edate	Expiration date	No
rdate	Date last referenced	No
migr	Whether the data set is migrated (YES or NO) based on the value of the VOLUME_OF_MIGRATED_DATA_SETS keyword in the ISPF configuration table. If the volume name of the data set matches the value of VOLUME_OF_MIGRATED_DATA_SETS, ZDLMIGR is set to YES, otherwise it is set to NO.	No
dsntp	Dsname type (PDS, LIBRARY, or ' ')	No
spacu	Space units	No
mvol	Whether the data set is multivolume (Y) or not (N)	No
ovf	Space overflow indicator (YES or NO)	No
dsorg	Data set organization	No
recfm	Record format	No
lrecl	Logical record length	No
blksz	Block size	No
size	Data set size in tracks, long format (12 bytes)	No
catnm	Name of the catalog where the data set is located	No
dev	Device type	NO

Data set list with attributes document

“Data set list with attributes document” on page 884 shows the attributes of the JSON data set list document.

For more information about these properties, see the dialog variables for the YES | NO | PRT parameter of **LMDLIST** (list data sets) in [LMDLIST-list data sets](#) in *z/OS ISPF Services Guide*.

Table 455. X-IBM-Attributes=vol

Property	Description	Required
items	An array where each element contains the following key:value pairs dsname Data set name. vol Volume in which the data set resides.	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes

Table 456. X-IBM-Attributes=dsname

Property	Description	Required
items	An array where each element contains the following key value pairs: dsname Data set name.	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	Optional property; set to true when more rows can be returned.	No
totalRows	Total number of rows that match the request.	No
JSONversion	Version number of this JSON document.	Yes

Table 457. X-IBM-Attributes=base

Property	Description	Required
items	An array where each element contains the following key value pairs: Table 458 on page 885	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes

Table 458. Items key:value pairs

Property	Description	Required
dsname	Data set name.	Yes

Table 458. Items key:value pairs (continued)		
Property	Description	Required
blksz	Block size.	No
catnm	Name of the catalog in which the data set was located.	No
cdate	Creation date.	No
dev	Device type.	No
dsntp	Dsname type (PDS, LIBRARY, or ' ') Percentage of used tracks or pages(PDSE).	No
dsorg	Data set organization.	No
edate	Expiration date.	No
extx	Number of extents used, long format (5 bytes).	No
lrecl	Logical record length.	No
migr	Whether the data set is migrated (YES or NO) based on the value of the VOLUME_OF_MIGRATED_DATA_SETS keyword in the ISPF configuration table. If the volume name of the data set matches the value of VOLUME_OF_MIGRATED_DATA_SETS, ZDLMIGR is set to YES, otherwise it is set to NO.	No
mvol	Whether the data set is multivolume (Y) or not (N).	No
ovf	Space overflow indicator (YES or NO).	No
rdate	Date last referenced.	No
recfm	Record format.	No
size	Data set size in tracks, long format (12 bytes).	No
spacu	Space units.	No
used	Percentage of used tracks or pages (PDSE).	No
vol	Volume serial.	No
vols	If mvol is Y , all the volumes are filled in this field,otherwise the value is equal to vol.	No

PDS/PDSE member list with attributes document

Contents of the [JSON PDS/PDSE member list document](#) shows the contents of the JSON member list document.

For more information about these properties, see the dialog variables for the YES|NO parameter description of **LMMFIND** (find a library member) in [LMMFIND—find a library member](#) in *z/OS ISPF Services Guide*.

Table 459. Contents of the JSON PDS/PDSE member list document

Property	Description	Required
items	An array where each element contains the following key:value pairs member Member name	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes

Table 460. X-IBM-Attributes=base and data set RECFM=F or V

Property	Description	Required
items	An array where each element contains the following key:value pairs. See Table 461 on page 887.	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes

Table 461. Items key:value pairs for data set RECFM=F or V

Property	Description	Required
member	member name.	Yes
vers	Version number; a number from 1 to 99.	No
mod	Modification level; a number from 0 to 99.	No
c4date	Creation date in 4-character year format	No
m4date	Last change date in 4-character year format	No
cnorc	Current number of records; a number from 0 to 65 535.	No
inorc	Beginning number of records; a number from 0 to 65 535.	No
mnorc	Number of changed records; a number from 0 to 65 535.	No

Table 461. Items key:value pairs for data set RECFM=F or V (continued)

Property	Description	Required
mtime	Last change time; a character value in the format hh:mm.	No
msec	Seconds value of the last change time. This is a two character field.	No
user	User ID of last user to change the given member; an alphanumeric field with a maximum length of 7 characters.	No
sclm	Indicates whether the member was last modified by SCLM or ISPF. A value of Y indicates the last update was made through SCLM. A value of N indicates that the last update was made.	No

Table 462. X-IBM-Attributes=base and data set RECFM=U

Property	Description	Required
items	An array where each element contains the following key:value pairs. See Table 463 on page 888 .	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
JSONversion	Version number of this JSON document.	Yes

Table 463. Items key:value pairs for data set RECFM=U

Property	Description	Required
member	member name.	Yes
ac	A 2-character field containing the authorization code of the member.	No
alias-of	An 8-character field containing the name of the real member that this member is an alias of. If the member is not an alias this field is blank.	No
amode	A 3-character field containing the AMODE of the member.	No

Table 463. Items key:value pairs for data set RECFM=U (continued)		
Property	Description	Required
attr	A 20-character field containing the load module attributes. The attributes are 2-character strings separated by blanks. These strings can appear in the attribute string: NX Not executable OL Only Loadable OV Overlay RF Refreshable RN Reentrant RU Reusable SC Scatter Load TS Test	No
rmode	A 3-character field containing the RMODE of the member.	No
size	An 8-character field containing the load module size in hex.	No
ttr	A 6-character field containing the TTR of the member.	No
ssi	An 8-character field containing the SSI information for a load module.	No

Create a sequential and partitioned data set document

Table 464 on page 889 shows the contents of the JSON sequential and partitioned data set document.

For more information about these properties, see the dialog variables for the YES | NO parameter description of **LMMFIND** (find a library member) in [LMMFIND—find a library member in z/OS ISPF Services Guide](#).

Table 464. Contents of the create a sequential and partitioned data set document		
Property	Description	Required
volser	Volume serial number of the device a data set will reside on.	No
unit	Unit name of the device that the data set will reside on.	No
dsorg	Unit name of the device that the data set will reside on.	No
alcunit	Unit of space allocation.	No
primary	Primary space allocation for the data set.	No
secondary	Secondary space allocation for the data set.	No
dirblk	Number of directory blocks for a partitioned data set.	No

Table 464. Contents of the create a sequential and partitioned data set document (continued)

Property	Description	Required
avgblk	Specifies the unit of space allocations to be blocks and sets the average block length.	No
recfm	Record format of a data set: The following formats are supported: <ul style="list-style-type: none"> • F: fixed • V: variable • U: undefined • B: block • FB: fixed blocked • VB: variable blocked 	No
blksize	Block size of the data set.	No
lrecl	Record length of data set.	No
storclass	Specifies the storage class of system managed storage.	No
mgntclass	Specifies the management class of the data set.	No
dataclass	Specifies the data class of the data set.	No
dsntype	Data set type. The following types are supported: <ul style="list-style-type: none"> • BASIC • EXTPREF • EXTREQ • HFS • LARGE • PDS • LIBRARY • PIPE 	No

File list document

Table 465 on page 890 shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service in z/OS ISPF Services Guide](#).

Table 465. Contents of the JSON file list document

Property	Description
items	JSON array of member entries containing the properties that follow in the remainder of this table.
filename	File name
returnedRows	Number of rows that were returned for this request.
moreRows	Optional property; set to true when more rows can be returned.
totalRows	Total number of rows that match the request.

Unix file and directory list with attributes document

Contents of the [unix file and directory](#) shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service](#) in *z/OS ISPF Services Guide*.

Table 466. Contents of the unix file and directory		
Property	Description	Required
items	An array where each element contains the following key:value pairs. See Unix items key:value pairs .	Yes
returnedRows	Number of rows that were returned for this request.	Yes
moreRows	True, if more rows are available to return.	No
totalRows	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	Yes
JSONversion	Version number of this JSON document	Yes

Table 467. Unix items key:value pairs		
Property	Description	Required
name	File or directory name.	Yes
mode	indicating the permissions.	No
size	For regular files, the file's size in bytes. For other kinds of files, the value of this field is unspecified.	No
uid	The numeric user ID (UID) of the file's owner.	No
user	The user name of the file's owner got by UID.	No
gid	The numeric group ID (GID) of the file's group.	No
group	The group name of the file's group got by GID.	No
mtime	The most recent time the contents of the file were changed.	No
target	If the file is symlink, this indicates the really file/directory	No

Mount and unmount a file system document

Contents of the [Mount and unmount a file system](#) document shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service](#) in *z/OS ISPF Services Guide*.

Table 468. Contents of the Mount and unmount a file system document

Property	Description	Required
action	Current support value: mount: you are going to mount an UNIX file system on the specified directory. unmount: you are going to unmount a specified directory.	Yes
mount-point	Specify the mount point you are going to mount/unmount, generally, it is directory.	No
fs-type	Specify the file system type you are going to mount, --the name must match the TYPE operand on a FILESYSTYPE statement in the BPXPRMxx parmlib member for the file system. For the mount action, this is required; but it is not required for the unmount action.	No
mode	Specify the mode you intend to mount the file system: Support Value: rdonly: read only rdwr: read write --all the values are case insensitive	No

Create a UNIX file document

Contents of the [create a UNIX file document](#) shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service](#) in *z/OS ISPF Services Guide*.

Table 469. Contents of the create a UNIX file document

Property	Description	Required
type	The request type, support value: directory or dir: you are going to create a directory. file: you are going to create a file.	No

Table 469. Contents of the create a UNIX file document (continued)		
Property	Description	Required
mode	<p>The mode specifies the file/directory permission bits to be used in creating the file/directory.</p> <p>rwXrwxrwx</p> <p>The 9 characters are in three groups of 3; they describe the permissions on the file. The first group of 3 describes owner permissions; the second describes group permissions; the third describes other (or "world") permissions. Characters that might appear are:</p> <p>r: Permission to read the file</p> <p>w: Permission to write on the file</p> <p>x: Permission to execute the file</p> <p>-: No such a permission</p>	No

Error report document

Table 470 on page 893 shows the contents of the JSON error report document for z/OS data set and file REST interface requests.

Table 470. Contents of the JSON error report document	
Property	Description
category	Error category. This field is integer data type.
rc	Return code. This field is integer data type.
reason	Reason code. This field is integer data type.
message	Message that describes the error.
details	(Optional) Array of strings containing additional message details.

For the meanings of the category, rc, and reason fields, see [“Error reporting categories” on page 893](#).

Error reporting categories

This section describes the error categories and associated error codes that can be returned in the JSON error report document for z/OS data set and file REST interface requests. This document is described in [“Error report document” on page 893](#).

Categories

Table 471 on page 894 shows the error categories that are defined for errors that are returned in z/OS data set and file REST interface operations.

Table 471. Error categories for z/OS data set and file REST interface operations

Category	Ordinal Value	Description	Where the error details are described
Service	1	Errors that are produced or detected in the service layer.	“Category 1 — Service error” on page 894
Message queue	2	Errors that are produced or detected by the message queue.	“Category 2 — Message queue error” on page 896
CEA	3	Errors that are produced or detected by the common event adapter (CEA) interface.	“Category 3 — Common event adapter (CEA) error” on page 897
ISPF	4	Errors that are produced or detected by the Interactive System Productivity Facility (ISPF) interface.	“Category 4 — ISPF error” on page 898
CSI	5	Errors that are produced or detected by the catalog search interface (CSI).	“Category 5 — Catalog Search Interface (CSI) error” on page 898
Read or write function	6	Errors that are returned from an attempted read or write request.	“Category 6 — Read or write error” on page 898
JSON	7	Errors that are produced or detected when parsing JSON.	“Category 7 — JSON Parser Conditions” on page 899
XL C/C++	8	Errors that are produced or detected by z/OS XL C/C++	“Category 8 — z/OS XL C/C++ Conditions” on page 900
Unexpected	16	Unexpected errors detected.	“Category 16 — Unexpected error” on page 900

Category 1 — Service error

Table 472 on page 894 shows the possible conditions for this error category.

Table 472. Category 1 errors for z/OS data set and file REST interface operations

rc	reason	message	Description
4	1	%s	The request specified a URL that is not valid. %s indicates which parts of the URL are invalid, such as path information, data set name and volume serial.
4	2	Invalid data set name length.	The request specified a data set name length that is not valid. Data set names cannot exceed 44 characters in length.
4	3	The specific data set name is invalid.	The request specified a data set name (dslevel) that is not valid.
4	4	Retrieving USS files should start with absolute path.	The requested file cannot be retrieved because the request is missing an absolute path name, which is required. When retrieving UNIX files, the file path must be an absolute (or fully qualified) path name, rather than a relative (or partially qualified) path name.

Table 472. Category 1 errors for z/OS data set and file REST interface operations (continued)

rc	reason	message	Description
4	5	Length of USS path is invalid.	The request specified a UNIX file path length that is not valid.
4	6	The specific path is invalid.	The request specified a UNIX file path that is not valid.
4	7	The specific volser is invalid.	The request specified a volume serial (volser) that is not valid.
4	8	File not found.	The request specified a file that does not exist.
4	9	Incorrect content type.	The request specified an unsupported content type.
4	10	Unrecognized HTTP method.	The request is not recognized as a supported HTTP method.
4	11	POST method is not supported.	The POST method was specified, however POST is not a supported HTTP method.
4	12	Incorrect attribute was specified.	The request contained one or more attributes that are not valid.
4	13	Incorrect parameter was specified.	The request contained one or more parameters that are not valid.
4	14	The specific JSON data in the request is invalid.	JSON data in the request is invalid.
4	15	The size of the data to be written is invalid.	The size of the data to be written is invalid.
4	16	Request content-length must be specified. Request content-length was too long. Request content was longer than specified content-length.	Only used in ValidateJsonServlet.C.
4	17	Request document contained invalid JSON.	Not currently used.
4	18	Member name is not valid.	Member name is not valid.
4	19	Unix file/directory exists.	UNIX file or directory already exists.
8	1	Unable to get JSON Response.	Not currently used.
8	2	Unable to get JSON Response Table.	Not currently used.
8	3	Unable to load ISPEXEC.	The ISPF service route ISPEXEC cannot be loaded.
8	4	Unable to load TSO servlet mappings.	The servlet dispatcher was not able to load the servlet mappings JSON file.
8	5	A servlet-mapping was not found for servlet-path.	A servlet-mapping matching the request cannot be found. Indicates a servlet-mapping configuration error.

Table 472. Category 1 errors for z/OS data set and file REST interface operations (continued)

rc	reason	message	Description
8	6	ExceptionServlet threw an exception after commit. Servlet failed but could not send error response.	An exception was thrown by the servlet.
8	7	TsoServlet already committed.	An exception was thrown by the servlet after its output stream header was committed.
8	9	Error executing command.	An error occurs when you use the z/OS UNIX file utilities.
16	1	Error occurred when connecting to remote server.	An error occurred when the X-IBM-Target-System header was set. See details to take action.

Category 2 – Message queue error

Table 473 on page 896 shows the possible conditions for this error category.

Table 473. Category 2 errors for z/OS data set and file REST interface operations

rc	reason	message	Description
4	1	Timeout	Timeout occurred when receiving a message from the message queue. MSG_QUEUE_PROTOCOL_ERROR_TIMEOUT
4	2	Received unexpected msgType=nn	Unexpected message received. MSG_QUEUE_PROTOCOL_ERROR_UNEXPECTED Note: If the message is login: received unexpected TSO LOGON, check whether the users set the header as X-IBM-Request-Acctnum.
4	3	ServletDispatcher failed.	Back-end servlet dispatcher failed with the reason <reason>. For example, TSO prompt was received when TsoServletResponse was expected. MSG_QUEUE_PROTOCOL_ERROR_ENDED
4	4	Queue full while sending.	Message queue is full while sending the specific <message type>. MSG_QUEUE_PROTOCOL_ERROR_FULL
4	5	Illegal state.	Message queue is in an invalid state. MSG_QUEUE_PROTOCOL_ERROR_ILLEGAL_STATE
5	1	Error parsing TsoServletResponse.	Error occurred when parsing TsoServletResponse. MSG_QUEUE_JSON_PARSE_ERROR
5	2	Error serializing TsoServletRequest.	Error occurred when serializing TsoServletRequest. MSG_QUEUE_JSON_SERIALIZE_ERROR

Table 473. Category 2 errors for z/OS data set and file REST interface operations (continued)

rc	reason	message	Description
12	1	The message queue cannot be created.	Message queue cannot be created.
12	2	Reserved.	Not used currently.
12	3	Message queue size is less than minimum.	Message queue size is less than the required minimum size. MSG_QUEUE_ERROR_SIZE_ERROR
12	4	Message prefix bytes are too short.	Message prefix bytes are shorter than expected. MSG_QUEUE_ERROR_DECODING
16	n(errno)	Varies.	Error from msgsnd(), msgrcv(), or a related message queue service. The UNIX errno is the reason code. For reason code 1141, the following causes are possible: <ul style="list-style-type: none"> • The TSO/E logon PROC is not correct. Ensure that the information specified on the RESTAPI_FILE statement in the IZUPRMxx parmlib member is correct. • The user ID does not have a TSO segment in the RACF user profile. • TSO/E logon exit is active and is preventing the user ID from logging on. MSG_QUEUE_SYS_ERROR

Category 3 – Common event adapter (CEA) error

Table 474 on page 897 shows the possible conditions for this error category.

Table 474. Category 3 errors for z/OS data set and file REST interface operations

rc	reason	message	Description
12	1	TSO launcher exception: Client is not authorized for instrumentation.	The requester lacks sufficient authority to access the requested common event adapter (CEA) service.
12	2	TSO launcher exception: Error occurred.	The requested CEA service encountered an error.
12	3	TSO launcher exception: CEA address space is not available.	The CEA address space is not active or is not available.
12	4	TSO launcher exception: TSO address space cannot be created.	The TSO/E address space cannot be created because a required system resource is not available.
12	5	CeaTsoEnd request failed.	An error occurred ending a TSO Address Space. CEA TSO Reason code = 4103. CEA_ERROR_END_TSO_FAILED

Table 474. Category 3 errors for z/OS data set and file REST interface operations (continued)			
rc	reason	message	Description
12	6		Not currently used. CEA_ERROR_CMD_NOT_FOUND
12	7	Unable to find the system you requested	CEA_ERROR_NO_TARGET_SYSTEM
12	n	Error occurs while using remote CEA TSO address space services	Refer to CEA TSO address space services reason code and diagnostic code for details.

Category 4 – ISPF error

Table 475 on page 898 shows the possible conditions for this error category.

Table 475. Category 4 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
n	m	varies	The return and reason code values match the return and reason code values that are set by the ISPF service.

Category 5 – Catalog Search Interface (CSI) error

Table 476 on page 898 shows the possible conditions for this error category.

Table 476. Category 5 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
n	m	varies	The return and reason code values match the return and reason code values that are set by the consolidated software inventory (CSI) service.

Category 6 – Read or write error

Table 477 on page 898 shows the return and reason codes that can be set for a read or write request.

Table 477. Category 6 errors for z/OS data set and file REST interface operations			
rc	reason (hex)	message	Description
8	201	<methodName> failed	Unable to open a data set or member, <methodName> like fopen() or freopen() failed. RW_ERROR_OPEN_FAILED
8	202	<methodName> failed	Unable to close a data set or member, <methodName> like fclose() failed. If this error occurs for a write or put operation, the data set contents are not predictable. failedRW_ERROR_CLOSE_FAILED

Table 477. Category 6 errors for z/OS data set and file REST interface operations (continued)

rc	reason (hex)	message	Description
8	204	Client ETag does not match the current ETag for the data set.	The attempt to write to the data set failed because the supplied ETag does not match the current ETag of the requested data set. This mismatch indicates that the data set content was modified in the time since the caller obtained the ETag. RW_ERROR_DS_ETAG_NOT_MATCHED
8	208	.<methodName> error	An error, <methodName> like fread(), occurred during I/O to a data set or UNIX file. RW_ERROR_IO
8	20C	Member not found.	The member cannot be located in the partitioned data set. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_MBR_NOT_FOUND
8	0000020A	Dynamic allocation failed.	This RC is combined with S99ERROR in the high halfword. RW_ERROR_DS_DYNALLOC_ERR
8	1708020A	ISPF LMINIT - data set not found.	The specified data set cannot be found. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_DS_NOT_FOUND
8	varies	Varies.	For UNIX file I/O errors, the reason code consists of the errno in the high order 16 bits and the errno2 in the low order 16 bits.
8	5E30062	File not found.	The specified UNIX file cannot be found. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_FS_NOT_FOUND
8	5B6F0002	Client is not authorized for file access.	The request for the UNIX file failed because the caller does not have sufficient authority to access the file. RW_ERROR_FS_AUTH
8	406	Client ETag does not match the current ETag for the file.	The attempt to write to the UNIX file failed because the supplied ETag does not match the current ETag of the requested file. This mismatch indicates that the file content was modified in the time since the caller obtained the ETag. RW_ERROR_FS_ETAG_NOT_MATCHED

Category 7 – JSON Parser Conditions

[Category 7 JSON parser conditions](#) shows the possible conditions for this error category.

Table 478. Category 7 JSON parser conditions			
rc	reason	message	Description
n	m	Varies.	The rc and reason are set from the low-level JSON Parser return code and reason code.

Category 8 – z/OS XL C/C++ Conditions

Category 8 z/OS XL C/C++ Conditions shows the possible conditions for this error category.

Table 479. Category 8 z/OS XL C/C++ Conditions			
rc	reason	message	Description
n	m	Varies.	<p>The rc and reason are set from the low-level z/OS XL C/C++ return code and reason code.</p> <p>If the message indicates a dynamic allocation error, then the rc is a decimal value and corresponds to a DYNALLOC reason code. For more information about the DYNALLOC reason code, see Interpreting error reason codes from DYNALLOC in <i>z/OS MVS Programming: Authorized Assembler Services Guide</i>.</p>

Category 16 – Unexpected error

Table 480 on page 900 shows the possible conditions for this error category.

Table 480. Category 9 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
16	1	Server error occurred.	For details about the exception, check the z/OSMF logs.
121	varies	iconv_open() failed	A possible reason for the error is that the header value X-IBM-Dsname-Encoding header is set but not valid.

z/OS jobs REST interface

The z/OS jobs REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with batch jobs on a z/OS system, as described in this topic.

Table 481 on page 900 lists the operations that the z/OS jobs REST interface services provide.

Table 481. Operations provided through the z/OS jobs REST interface services.	
Operation	HTTP method and URI path
“Obtain the status of a job” on page 905	GET /zosmf/restjobs/jobs/<jobname>/<jobid>?[step-data=Y N] GET /zosmf/restjobs/jobs/<correlator>?[step-data=Y N]
“List the jobs for an owner, prefix, or job ID” on page 909	GET /zosmf/restjobs/jobs[?<parms>]
“List the spool files for a job” on page 912	GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files GET /zosmf/restjobs/jobs/<correlator>/files

Table 481. Operations provided through the z/OS jobs REST interface services. (continued)

Operation	HTTP method and URI path
<u>“Retrieve the contents of a job spool file” on page 914</u>	GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/<nnn>/records GET /zosmf/restjobs/jobs/<correlator>/files/<nnn>/records GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/JCL/records GET /zosmf/restjobs/jobs/<correlator>/files/JCL/records
<u>“Submit a job” on page 918</u>	PUT /zosmf/restjobs/jobs[/-<JESB>]
<u>“Hold a job” on page 923</u>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<u>“Release a job” on page 926</u>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<u>“Change the job class” on page 928</u>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<u>“Cancel a job” on page 931</u>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<u>“Cancel a job and purge its output” on page 933</u>	DELETE /zosmf/restjobs/jobs/<jobname>/<jobid> DELETE /zosmf/restjobs/jobs/<correlator>

Using the Swagger interface

You can use the Swagger interface to display information about the z/OS jobs REST APIs. The Swagger interface includes one section: Jobs APIs. For more information, see [“Using the Swagger interface” on page 50](#).

Processing overview

The z/OS jobs REST interface services can be invoked by any HTTP client application, running on the z/OS local system or a remote system.

Your program (the client) initiates an HTTP request to the z/OS jobs REST interface. If the interface determines that the request is valid, it performs the requested service. After performing the service, the z/OS jobs REST interface creates an HTTP response. If the request is successful, this response takes the form of an HTTP 2nn response and, if applicable, a result set that is passed back to your program.

Depending on which service was requested, the result set might be returned in a format that requires parsing by your program, for example, a JSON object. In other cases, results might be returned in another format, such as plain text or binary data. If the request is not successful, the response consists of a non-OK HTTP response code with details of the error that is provided in the form of a JSON object. The contents of the JSON objects are described in [“JSON document specifications for z/OS jobs REST interface requests” on page 936](#).

Resource URLs

The URLs of the z/OS jobs REST interface have the format that is shown in [Figure 425 on page 902](#):

```
https://{host}:{port}/zosmf/restjobs/jobs/{jesb}/{resource}?{parm}
```

Figure 425. Format of resource URLs for z/OS jobs REST interface.

Where:

- "https://{host}:{port}" specifies the target system address and port.
- "/zosmf/restjobs/jobs/" identifies the z/OS jobs REST interface.
- "JESB" optionally specifies a secondary JES subsystem, if one is to be used to process the request. If you omit this value, the request is processed by the primary JES subsystem.
- "{resource}?{parm}" represents the resource, such as a job name and job ID, and optionally one or more parameters, to qualify the request.

HTTP methods

The z/OS jobs REST interface provides the following HTTP methods:

GET

Retrieves information about jobs that are running on the z/OS system.

PUT

Updates job information on the z/OS system, or sets attributes and performs actions on jobs.

DELETE

Removes jobs from the z/OS system.

Some situations might require the use of the POST method; see [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Supported HTTP versions

z/OS jobs REST interface supports requests in either of the following protocols: HTTP/1.0 or HTTP/1.1

Content types

The data that is sent or returned by the HTTP methods has one of the following content types:

- Application/octet-stream (Content-Type: application/octet-stream) is used for data that is sent or returned in an uninterpreted format, such as a job that is submitted, or binary data or records that are obtained from a z/OS job spool file.
- JSON (Content-Type: application/json) is used for sent data and returned data. For the detailed format of each JSON object, see the description for each operation.
- Plain text (Content-Type: plain/text).

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the problem or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document”](#) on page 942.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 201 Created

Request was successful, and, as a result, a resource was created.

HTTP 202 Accepted

Request was received and accepted for processing, but the processing has not yet completed.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 500 Internal server error

Programming error.

Synchronous support for the job modify operations

The z/OS jobs REST interface includes services that you can use to perform job modify operations, as shown in [Table 482 on page 903](#).

<i>Table 482. Job modify operations provided through the z/OS jobs REST interface services.</i>	
Operation	Where described
Hold a job.	“Hold a job” on page 923
Release a job.	“Release a job” on page 926
Change the job class.	“Change the job class” on page 928
Cancel a job.	“Cancel a job” on page 931
Delete a job (cancel a job and purge its output).	“Cancel a job and purge its output” on page 933

These services can be run synchronously if coded to use the latest version of the service. To request synchronous processing, set the "version" property in your request to 2.0 or omit the "version" property. If so, the system attempts to process the request synchronously, if such processing is supported on the target JES subsystem. Synchronous processing is supported for JES2 subsystems only. When the target subsystem is JES3, a synchronous request is ignored and the service is performed asynchronously.

Generally, the differences in processing are as follows:

- For an asynchronous request, z/OSMF returns control to the caller immediately. However, to verify that the initial request was performed, the caller must then issue the service that is described in [“Obtain the status of a job” on page 905](#).
- For a synchronous request, z/OSMF does not return control to the caller until the requested action is performed and results of the request are available to be returned to the caller. Here, the JSON job feedback document provides more information about the success or failure of the request; see [“Job feedback document” on page 940](#).

If your program does not require feedback on the results of requested actions, you can use these services asynchronously.

Due to timing behavior, if you submit a job and immediately issue a synchronous request for the same job, you might receive the error message "No job found for reference" in the JSON error report document (category 6, return code 4, reason code 10). To avoid this occurrence, it is recommended that you allow a small amount of time to pass between a job submit request and a subsequent job modify request.

Required system setup

The z/OS jobs REST interface services require that the System REXX (SYSREXX) component is set up and active on your z/OS system. For information, see [Ensuring that System REXX is set up and active in IBM z/OS Management Facility Configuration Guide](#).

Required authorizations

Generally, your user ID requires the same authorizations for using the z/OS jobs REST interface services as when you perform these operations through a TSO/E session on the system. For example, submitting a job through the z/OS jobs REST interface requires that your user ID is authorized to run jobs on the system and can access any protected resources that the job might require.

In addition, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <SAF_PREFIX> in the APPL class
- READ access to the <SAF_PREFIX>.*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF_PREFIX>.IzuManagementFacilityRestJobs resource name in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

If you are using z/OS jobs REST interface services to perform job modify operations, your user ID must be authorized to the appropriate resources in the JESJOBS class, as shown in Table 483 on page 904.

Table 483. JESJOBS class authorizations needed for performing job modify operations		
Operation	JESJOBS resource	Access required
Hold a job	HOLD.nodename.userid.jobname	UPDATE
Release a job	RELEASE.nodename.userid.jobname	UPDATE
Change the job class	MODIFY.nodename.userid.jobname	UPDATE
Cancel a job	CANCEL.nodename.userid.jobname	ALTER
Delete a job (cancel a job and purge its output)	CANCEL.nodename.userid.jobname	ALTER

For more information about JESJOBS class, see [Controlling the use of job names in z/OS Security Server RACF Security Administrator's Guide](#).

If run asynchronously, these services also require that your user ID is authorized to the Common Information Model (CIM) server and permitted to the JES2-JES3Jobs CIM provider. CIM includes jobs (CFZSEC and CFZRCUST) to help you configure the CIM server, including security authorizations and file system customization. For information, see [Quick guide: CIM server setup and verification in z/OS Common Information Model User's Guide](#).

Where applicable, additional authorization requirements are noted in the descriptions of the individual z/OS jobs REST interface services. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Requesting asynchronous job notifications

You can use the asynchronous job notifications function of z/OSMF to allow your programs to be notified of job events.

The asynchronous job notifications function is available for JES2 systems only; it is not available for JES3 systems.

There are two mechanisms to support asynchronous job notifications:

- JES2 EDS for job notifications over HTTP
- Common Information Model (CIM) jobs indication provider

With this function, the program that submits the job through the z/OS jobs REST interface services PUT method specifies a URL when submitting the job. When using JES2 EDS, you can define the events of a submitted job, and if any of those events happen, z/OSMF returns an HTTP message to the URL location.

When using CIM, z/OSMF returns an HTTP message to the URL location only when a job ends, indicating the job completion status. The data returned is in the form of a JSON document.

You only need to configure one of the mechanisms and it is recommended you use JES2 EDS for job notifications over HTTP. For instructions on enabling the asynchronous job notifications function, see [Configuring your system for asynchronous job notifications in IBM z/OS Management Facility Configuration Guide](#).

Enabling browser login through a client certificate

It is possible to run the z/OS jobs REST interface services directly from a web browser. Here, you must first authenticate to z/OSMF through your browser. In z/OSMF authentication is typically done by entering your user ID and password at the **Welcome** page. However, it is also possible to log in with a client certificate, if your installation favors this approach. With a client certificate, you can access z/OSMF through your browser without having to supply a user ID and password.

In client certificate authentication, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

If z/OSMF does not accept your client certificate, z/OSMF displays the Welcome page for you to enter your user ID and password.

If your installation plans to enable client certificate login for the z/OS jobs REST interface services, understand that it is your responsibility to create the certificate and manage its distribution to users. It is recommended that you ensure that users have browsers that support importing a certificate.

For information about creating digital certificates, see [RACF and digital certificates](#) in *z/OS Security Server RACF Security Administrator's Guide*.

Error logging

Errors from the z/OS jobs REST interface services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Enabling traces for IBM analysis

For diagnostic purposes, your installation might be asked by IBM Support to enable tracing for the z/OS jobs REST interface. For information, see [Appendix A, “Enabling tracing for the z/OS jobs REST interface,” on page 1283](#)

Obtain the status of a job

You can use this operation to obtain the status of a batch job on z/OS.

HTTP method and URI path

```
GET /zosmf/restjobs/jobs/<jobname>/<jobid>?[step-data=Y|N]
GET /zosmf/restjobs/jobs/<correlator>?[step-data=Y|N]
```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:

- **<jobname>/<jobid>** identifies the job for which status is requested. **<jobname>** can contain a question mark (?) that represents a single character.
- **<correlator>** identifies the job for which status is requested. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **[step-data]** is an optional parameter that indicates whether the service returns information about each step in the job that completed, such as the step name, step number, and completion code. To return the step data for completed steps, set the value for this parameter to Y. Otherwise, omit the parameter or set its value to N (default). If a step is run multiple times, duplicate data for that step might be included in the response.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

You can specify following optional query parameters on this request:

user-correlator

The user portion of the job correlator. This value is 1-32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

This query parameter is mutually exclusive with jobid.

This value is processed by the JES2 subsystem only; the JES3 subsystem does not process the correlator and instead, indicates zero job matches. For a system with JES3 as the primary subsystem, and one or more JES2 secondary subsystems, the primary JES3 subsystem does not process the correlator. However, the JES2 secondary subsystems can process the correlator.

exec-data

This optional parameter specifies whether to return execution data about the job, if execution data is available. This parameter is a string value and is case-insensitive. Valid values are:

- Y (or y)
- N (or n)

For example: `exec-data=Y`

The following execution data is returned for each job:

exec-system

System name of the z/OS system on which the job ran (up to 8 characters)

exec-member

Member name of the z/OS system on which the job ran (up to 8 characters)

exec-submitted

Time when the job was submitted to run (the input end time)

exec-started

Time when job execution started

exec-ended

Time when job execution ended.

Timestamps are presented in the JSON UTC format: `"yyyy-mm-ddThh:mm:ss.mmmZ"`

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark (?) to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).
- Wildcard characters are permitted in the owner and prefix query parameter values. Use an asterisk (*) for multiple characters, and a question mark (?) for a single character.

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with a JSON job document. For the contents, see [“Job document” on page 936](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

The following request obtains the status for the job BLSJPRMI, job ID STC00052:

```
GET /zosmf/restjobs/jobs/BLSJPRMI/STC00052?exec-data=Y HTTP/1.1
Host: zosmf1.yourco.com
```

Example responses

A sample response is shown in [Figure 426 on page 908](#).

```
HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:06:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "STC00052",
  "jobname": "BLSJPRMI",
  "subsystem": "JES2",
  "owner": "IBMUSER",
  "status": "OUTPUT",
  "type": "STC",
  "class": "STC",
  "retcode": "CC 0000",
  "url": "https://host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A",
  "files-url": "https://host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A/files",
  "job-correlator": "S0000052SY1.....CE35BDE8.....:",
  "phase": 20,
  "phase-name": "Job is on the hard copy queue",
  "exec-system": "SY1",
  "exec-member": "SY1",
  "exec-submitted": "2018-11-03T09:05:15.000Z",
  "exec-started": "2018-11-03T09:05:18.010Z",
  "exec-ended": "2018-11-03T09:05:25.332Z"
}
```

Figure 426. Example: Returned job status

A sample response for an active step is shown in [Figure 427 on page 908](#).

```
HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:06:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "retcode": "null",
  "jobname": "BLSJPRMI",
  "status": "ACTIVE",
  "job-correlator": "S0000052SY1.....CE35BDE8.....:",
  "class": "STC",
  "type": "STC",
  "jobid": "STC00052",
  "url": "https://host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A",
  "phase-name": "Job is on the hard copy queue",
  "step-data": [
    {
      "smfid": "SP21",
      "active": true,
      "step-number": 1,
      "proc-step-name": "STARTING",
      "step-name": "IEFPROC",
      "program-name": "BLSQPRMI"
    }
  ],
  "owner": "IBMUSER",
  "subsystem": "JES2",
  "files-url": "https://host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A/files",
  "phase": 20,
  "exec-system": "SY1",
  "exec-member": "SY1",
  "exec-submitted": "2018-11-03T09:05:15.000Z",
  "exec-started": "2018-11-03T09:05:18.010Z",
  "exec-ended": "2018-11-03T09:05:25.332Z"
}
```

Figure 427. Example: Returned status for an active step

List the jobs for an owner, prefix, or job ID

You can use this operation to list the jobs for an owner, prefix, or job ID.

HTTP method and URI path

```
GET /zosmf/restjobs/jobs[?<parms>]
```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- **<parms>** are optional parameters that you can use to qualify the request. For a list of the supported parameters, see [“Query parameters” on page 909](#).

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

You can specify one or more of the following optional query parameters on this request:

owner

User ID of the job owner whose jobs are being queried; the default is the z/OS user ID. Folded to uppercase; cannot exceed 8 characters.

prefix

Job name prefix; default is *. Folded to uppercase; cannot exceed 8 characters.

jobid

Job ID. Folded to uppercase; cannot exceed 8 characters. This query parameter is mutually exclusive with user-correlator.

max-jobs

Maximum number of jobs returned. The value must be in the range 1 - 1000. If this parameter is not specified, or is specified incorrectly, the default value of 1000 is used.

user-correlator

The user portion of the job correlator. This value is 1 – 32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

This query parameter is mutually exclusive with `jobid`.

This value is processed by the JES2 subsystem only; the JES3 subsystem does not process the correlator and instead indicates zero job matches. For a system with JES3 as the primary subsystem, and one or more JES2 secondary subsystems, the primary JES3 subsystem does not process the correlator, however, the JES2 secondary subsystems can process the correlator.

exec-data

This optional parameter specifies whether to return execution data about the job, if execution data is available. This parameter is a string value and is case-insensitive. Valid values are:

- Y (or y)
- N (or n)

For example: `exec-data=Y`

The following execution data is returned for each job:

exec-system

System name of the z/OS system on which the job ran (up to 8 characters)

exec-member

Member name of the z/OS system on which the job ran (up to 8 characters)

exec-submitted

Time when the job was submitted to run (the input end time)

exec-started

Time when job execution started

exec-ended

Time when job execution ended.

Timestamps are presented in the JSON UTC format: `"yyyy-mm-ddThh:mm:ss.mmmZ"`

By default, execution data for up to 100 jobs is returned. If this parameter is specified, its value overrides the `max-jobs` value, which defaults to 1000. If you set `max-jobs` to less than 100, the `max-jobs` value is used. Otherwise, the default of 100 is used.

status

You can use this optional parameter to limit the returned data to only jobs that are currently active. This parameter is case-insensitive.

The following values are valid:

- `status=ACTIVE` or `status=active`

If you omit this parameter, the response includes data for both active and inactive jobs.

If you set this parameter to an incorrect value, the parameter is ignored and the default is used.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark (?) to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).
- Wildcard characters are permitted in the owner and prefix query parameter values. Use an asterisk (*) for multiple characters, and a question mark (?) for a single character.

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, observe the following considerations for this request:

- The ordering of the jobs that are returned is not predictable.
- If the maximum number of jobs is returned, no indication is provided for whether more jobs remain to be retrieved.
- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB  
https://host:port/zosmf/restjobs/jobs/-JESB?owner=owner
```

where *JESB* is the name of the secondary JES subsystem in these examples.

- To list the jobs for a job ID on a secondary JES subsystem, you must specify the job ID, rather than a job correlator. For example:

```
https://host:port/zosmf/restjobs/jobs/-JESB?jobid=jobid
```

Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with an array of matching jobs, each as a JSON job document. For the contents, see [“Job document” on page 936](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

In the following example, the GET method is used to list the jobs that are owned by IBMUSER and have a job name prefix that begins with TESTJOB:

```
GET /zosmf/restjobs/jobs?owner=IBMUSER&prefix=TESTJOB*&exec-data=Y  
Host: zosmf1.yourco.com
```

Example response

A sample response is shown in [Figure 428 on page 912](#).

```

HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:07:12 +0000GMT
Content-Type: application/json
Connection: close

[
  {
    "jobid": "JOB00023", "jobname": "TESTJOB2", "subsystem": null, "owner": "IBMUSER",
    "status": "OUTPUT", "type": "JOB", "class": "A", "retcode": "CC 0000",
    "url": "https://host:port/zosmf/restjobs/jobs/TESTJOB2/JOB00023",
    "files-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB2/JOB00023/files",
    "exec-system": "SY1", "exec-member": "SY1", "exec-submitted": "2018-11-03T09:05:15.000Z",
    "exec-started": "2018-11-03T09:05:18.010Z", "exec-ended": "2018-11-03T09:05:25.332Z",
    {
      "jobid": "JOB00024", "jobname": "TESTJOB3", "subsystem": null, "owner": "IBMUSER",
      "status": "OUTPUT", "type": "JOB", "class": "A", "retcode": "ABEND S000",
      "url": "https://host:port/zosmf/restjobs/jobs/TESTJOB3/JOB00024",
      "files-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB3/JOB00024/files",
      "exec-system": "SY1", "exec-member": "SY1", "exec-submitted": "2018-11-03T09:06:30.000Z",
      "exec-started": "2018-11-03T09:06:36.020Z", "exec-ended": "2018-11-03T09:06:50.007Z"
    }
  ]

```

Figure 428. Example: Returned list of the jobs for a specific owner and job name prefix

List the spool files for a job

You can use this operation to list the spool files for a batch job on z/OS.

HTTP method and URI path

```

GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files
GET /zosmf/restjobs/jobs/<correlator>/files

```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job for which the spool files are to be listed. **<jobname>** can contain a question mark (?) that represents a single character.
 - **<correlator>** identifies the job for which the spool files are to be listed. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **/files** indicates that the response is to list the spool files for the specified job.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request

- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, note that this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid/files
```

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with an array of zero or more JSON job file documents. For the contents, see [“Job file document” on page 940](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

The following request lists the spool files for the job TESTJOB1, job ID JOB00023:

```
GET /zosmf/restjobs/jobs/TESTJOB1/JOB00023/files HTTP/1.1
```

Example response

A sample response is shown in [Figure 429 on page 914](#).

```

HTTP/1.1 200 OK
Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

[
  {
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 1,
    "stepname": "JESE", "procstep": null, "class": "H",
    "ddname": "JESMSGLG", "record-count": 14, "byte-count": 1200,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/1/records",
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 2,
    "stepname": "JESE", "procstep": null, "class": "H",
    "ddname": "JESJCL", "record-count": 10, "byte-count": 526,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/2/records",
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 3,
    "stepname": "JESE", "procstep": null, "class": "H",
    "ddname": "JESYSMSG", "record-count": 14, "byte-count": 1255,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/3/records",
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 4,
    "stepname": "STEP57", "procstep": "COMPILE", "class": "H",
    "ddname": "SYSUT1", "record-count": 6, "byte-count": 741,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/4/records",
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 5,
    "stepname": "STEP57", "procstep": "COMPILE", "class": "A",
    "ddname": "SYSPRINT", "record-count": 3, "byte-count": 209,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/5/records"
  }
]

```

Figure 429. Example: Returned list of spool files

Retrieve the contents of a job spool file

You can use this operation to retrieve the contents of a job spool file on z/OS. Also, you can use this service to retrieve the JCL that was used to submit the job.

HTTP method and URI path

```

GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/<nnn>/records
GET /zosmf/restjobs/jobs/<correlator>/files/<nnn>/records
GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/JCL/records
GET /zosmf/restjobs/jobs/<correlator>/files/JCL/records

```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job for which the spool file contents are requested. **<jobname>** can contain a question mark (?) that represents a single character.
 - **<correlator>** identifies the job for which the spool file contents are requested. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **/files<nnn>/records** indicates that the request is to retrieve the contents of a job spool file for the specified job. The **<nnn>** parameter is the ID for the spool file from which the contents are to be retrieved.
- **/files/JCL/records** indicates that the request is to retrieve the JCL for the specified job.

Custom headers

You can include the following optional custom HTTP header with this request:

X-IBM-Record-Range

Use this header to retrieve a range of records from a spool file. You can specify this range by using either of the following formats:

SSS-EEE

Where *SSS* identifies the start record and *EEE* identifies the end record to be retrieved. Both values are relative offsets (0-based). When *EEE* is set to 0, records through the end of the spool file are retrieved.

SSS,NNN

Where *SSS* identifies the start record and *NNN* identifies the number of records to be retrieved.

For an example of how this custom header is used, see [“Examples” on page 917](#).

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

You can specify one or more of the following optional query parameters on this request.

mode

Use the mode parameter to specify conversion options for the returned data. The following values are valid for mode:

text

The z/OS jobs REST interface converts records from the server code page to the client code page and returns the records with Content-Type: plain/text. Trailing spaces are removed and newline characters are used as record separators. This value is the default if you omit the mode parameter.

binary

The z/OS jobs REST interface performs no conversion and returns the records with Content-Type: application/octet-stream.

record

The z/OS jobs REST interface performs no conversion and returns the records with Content-Type: application/octet-stream. The z/OS jobs REST interface prefixes each record with a 4-byte (big endian) length.

Specifying the mode parameter with any other value, or no value, results in the default: mode=text.

When mode=text, the following query parameters can be used to further qualify the request. These parameters cannot be used when mode is set to record or binary; doing so results in an error.

- fileEncoding
- search

- research
- insensitive
- maxreturnsize

These query parameters are described as follows.

fileEncoding=code-page

Specifies an alternative code page (EBCDIC) for the spool file on z/OS; the encoded text is converted to the client's request encoding. If not specified, the default code page is IBM-1047.

search=<string>

The spool file is searched for the first record that contains the string, without respect to case (by default). Optionally, insensitive=false can be specified for case-sensitive matching.

This parameter cannot be used with the research parameter.

research=<regular-expression>

The spool file is searched for the first record that matches the given extended regular expression. For example, research=A|B finds A or B. By default, the search is case-insensitive. To search for case-sensitive matches, specify the research parameter with the query parameter insensitive=false.

This parameter cannot be used with the search parameter.

insensitive=true|false

When 'true', searches (search and research) are case-insensitive. For case-sensitive searches, specify 'false'. The default is 'true'.

maxreturnsize=<integer>

This parameter can be specified only with search= or research=.

The value given is the maximum number of records to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching record. The X-IBM-Record-Range request header can be used to specify the range of records to be searched, but it does not restrict the number of records returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search begins with the first record. In all cases, an X-IBM-Record-Range=p,q response header is returned, where p is the first matching record and q is the number of records returned. If no matching records are found, the response header X-IBM-Record-Range=0,0 is returned.

The parameter cannot be used if the mode query parameter specifies any option except 'text'.

Required authorizations

See [“Required authorizations” on page 904](#).

In addition, your user ID requires READ access to the JESSPOOL profile for the spool data set. If no profile exists, only the user who created the spool data set can access, modify, or delete it. For information about spool data set security considerations, see [z/OS JES Application Programming](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, observe the following considerations for this request:

- The response does not include the Content-Length header. Because the server streams the data rather than buffering it in memory, the server cannot determine the total content length of the data before it completes the transfer. For similar reasons, the response does not include the Content-Range header, either.
- This request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid/filesJCL/records
```

Where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with content type that is defined by the mode query parameter.

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 942](#).

Examples

In the following example, the GET method is used to retrieve the contents of spool file 1 for the job TESTJOB, job ID JOB00023:

```
GET /zosmf/restjobs/jobs/TESTJOB/JOB00023/files/1/records HTTP/1.1
Host: zosmf1.yourco.com
```

A sample response is shown in [Figure 430 on page 917](#).

```
HTTP/1.1 200 OK
Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: text/plain
Connection: close

      J E S 2 J O B L O G -- S Y S T E M E I M G -- N O D E D C E I M G W V

15.49.11 JOB00239 ---- MONDAY, 14 JAN 2013 ----
15.49.11 JOB00239 IRR010I USERID IBMUSER IS ASSIGNED TO THIS JOB.
15.49.11 JOB00239 ICH70001I IBMUSER LAST ACCESS AT 15:48:25 ON MONDAY, JANUARY 14, 2013
15.49.11 JOB00239 $HASP373 INSTALL STARTED - INIT 2 - CLASS A - SYS EIMG
15.49.11 JOB00239 IEF403I INSTALL - STARTED - TIME=15.49.11
15.49.16 JOB00239 IEF404I INSTALL - ENDED - TIME=15.49.16
15.49.16 JOB00239 $HASP395 INSTALL ENDED
----- JES2 JOB STATISTICS -----
14 JAN 2013 JOB EXECUTION DATE
      71 CARDS READ
      287 SYSOUT PRINT RECORDS
        0 SYSOUT PUNCH RECORDS
        13 SYSOUT SPOOL KBYTES
      0.08 MINUTES EXECUTION TIME
```

Figure 430. Example: Returned spool file content

In the following example, the GET method is used to retrieve a range of records (the first 250) using the X-IBM-Record-Range custom header:

```
GET /zosmf/restjobs/jobs/TESTJOB/JOB00023/files/8/records HTTP/1.1
X-IBM-Record-Range: 0-249
```

A sample response is shown in [Figure 431 on page 918](#).

```
HTTP/1.1 200 OK
Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: text/plain
Connection: close
...(the first 250 records)
```

Figure 431. Example: Returned spool file content (a range of records)

In the following example, the GET method is used to retrieve the JCL for the job TESTJOB, job ID JOB00060:

```
GET /zosmf/restjobs/jobs/TESTJOB/JOB00060/files/JCL/records HTTP/1.1
```

A sample response is shown in [Figure 432 on page 918](#).

```
HTTP/1.1 200 OK
//TESTJOB JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14
```

Figure 432. Example: Returned job content (the job JCL)

Submit a job

You can use this operation to submit a job to run on z/OS.

HTTP method and URI path

```
PUT /zosmf/restjobs/jobs[/-<JESB>]
```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- **<JESB>** represents an optionally specified secondary JES subsystem. If you omit this value, the request is processed by the primary JES subsystem.

Standard headers

Use the following standard HTTP header with this request:

Content-Type

One of the following values:

- Set to `text/plain` when the optional header `X-IBM-Intrdr-Mode` is set to `TEXT` or is omitted, and the job JCL is included in the request.
- Set to `application/octet-stream` when optional header `X-IBM-Intrdr-Mode` is set to `RECORD` or `BINARY`, and the JCL for the job to be submitted is included in the HTTP request.
- Set to `application/json` when the job to be submitted resides in a data set or UNIX file, which is identified in a JSON document (included as input with this request).

Custom headers

Optionally, you can include one of the following custom HTTP headers with this request:

X-IBM-Intrdr-Class

A single character that specifies the internal reader class; the default is A. This value defines the default message class (MSGCLASS) for the job.

X-IBM-Intrdr-Recfm

A single character that specifies the internal reader record format: F or V.

When submitting a job from a data set, you can omit this header. Otherwise, this value must match the record format of the data set.

When not submitting a job from a data set, if you omit this header or specify a value other than F or V, the default of F is used.

X-IBM-Intrdr-Lrecl

An integer value that specifies the internal reader logical record length (LRECL).

When submitting a job from a data set, you can omit this header. Otherwise, this value must match the LRECL of the data set.

When not submitting a job from a data set, if you omit this header or specify a non-integer value, the default of 80 is used.

X-IBM-Intrdr-Mode

A keyword that specifies the format of the input job: TEXT, RECORD, or BINARY.

When submitting a job from a data set, you can omit this header. Otherwise, this value must be set to RECORD.

When not submitting a job from a data set, observe the following rules:

- If you omit this header, the TEXT keyword is used by default.
- If you specify the BINARY keyword, the X-IBM-Intrdr-Recfm header must be omitted or set to F (the default).
- If you specify the RECORD keyword or BINARY keyword, you must set Content-Type to `application/octet-stream`.

X-IBM-User-Correlator

Specifies the user portion of the job correlator. In z/OS, a job correlator can be used to associate each job with a unique 64-character value. The correlator provides you with a means to query a job in the system and track it through execution.

A job correlator consists of a 31-byte system-defined portion and a colon character (:), followed by a 32-byte user portion. The system-defined portion contains the following values:

- 8-byte job ID
- 8-byte MAS name for the system on which the job resides
- 8-byte sequence value
- 7-bytes of reserved space.

Following the system value is the colon character (:) separator and the second string: an optional 32-character user-defined value (the user portion). This value is 1 – 32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

If specified, the user portion is combined with the system portion, producing the full job correlator that will be returned in the job-correlator property of the JSON job document. If the user portion is not specified, the returned job correlator is the 32-byte system value, ending with the colon (:).

If this header is specified when JES3 is the primary job entry subsystem, an error results and no job is submitted.

For more information on the job correlator, see *z/OS JES2 Commands*.

X-IBM-JCL-Symbol-name

Specifies the name and value for a JCL symbol. The symbol name is included in the header, at the name position. The characters that follow 'X-IBM-JCL-Symbol-' up to the colon separator (:) form the symbol name. The data that follows the colon specifies the value for the symbol.

A symbol name is 1–8 characters, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 7) can be any combination of uppercase alphabetic, numeric (0-9), or special.

A symbol value is limited to 255 characters. Multiple symbol names and values can be specified, up to a limit of 128.

Example: X-IBM-JCL-Symbol-MBR: ABC specifies symbol name MBR with value ABC. Specifying this custom header and submitting a job that uses //MYDD DSN=MY.DATASET(&MBR.) , DISP=SHR in the JCL will cause ABC to be substituted as the member name.

If this header is specified when JES3 is the primary job entry subsystem, an error results and no job is submitted.

For more information on JCL symbols, see [Using system symbols and JCL symbols in z/OS MVS JCL Reference](#).

X-IBM-Notification-URL

Specifies a destination URL for receiving an HTTP POST when any of the job events specified in X-IBM-Notification-Options happen. If X-IBM-Notification-Options is not specified, the destination URL receives an HTTP POST when the job is no longer eligible for execution (that is, when the job reaches the output queue or purge queue). The notification is in the form of a JSON document (Content-Type: application/json), which contains job status information. For the contents of the JSON document, see [“Job notification document”](#) on page 938.

X-IBM-Notification-Options

Specifies the events of a submitted job. This header consists of a JSON string with a single property called events. The value of the events property is a list of strings that indicates the job events which trigger HTTP POSTs. If any of the job events happen, the destination URL specified in X-IBM-Notification-URL receives an HTTP POST. Valid event types are ready, active, and complete and are case-insensitive. The maximum number of characters is 4096. If no values or invalid event types are provided, an exception occurs.

- Ready - Job is ready for execution.
- Active - Job started execution.
- Complete - Job completed execution.

This header is only valid if you are using JES2 EDS for job notifications over HTTP. It will not take effect if you are using the Common Information Model (CIM) jobs indication provider.

If this header is specified, the header X-IBM-Notification-URL must also be specified, otherwise it is ignored.

Example:

```
X-IBM-Notification-Options: {"events": ["active", "ready", "complete"]}
```

X-IBM-Intrdr-File-Encoding

This optional header specifies that the EBCDIC code page is to be used for encoding the data that is written to the internal reader. If not specified, the default is IBM-1047. This header is ignored when optional header X-IBM-Intrdr-Mode is set to RECORD or BINARY.

This header is effective only when the JCL for the job to be submitted is included in the HTTP request body. This header is ignored when the job to be submitted resides in a data set or UNIX file.

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-

System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Input to this request

- Internet media type: text/plain, application/octet-stream, or application/json
- HTTP request with optional headers, followed by job to be submitted or a JSON document identifying the location of the job to be submitted (a data set or UNIX file).

To submit a job, you can include the job JCL in the HTTP request itself, or you can have the request refer to a job that resides in a data set or UNIX file. Here, you include a JSON document ("Content-Type: application/json" with the HTTP request. The JSON document contains the property "file": "<file-name>" where <file-name> identifies the data set or UNIX file that contains the job to be submitted.

Use the JSON document to identify the data set or UNIX file containing the job to be submitted, as follows:

- For a data set, specify the qualified data set name, prefixing the data set name with two leading forward slash characters ("//").

If not fully qualified, the current z/OSMF user ID is prefixed to the data set name. Supported data set types include sequential data sets and members of partitioned data sets.

Data sets must be catalogued.

- For a z/OS UNIX file, specify the absolute path name of the file.

Code page conversion is not performed on the contents of the file.

For a migrated data set, this operation does not cause the data set to be retrieved, unless you specify otherwise. To request that a data set be recalled without waiting, you can specify the "recall" property with the value "yesnowait" to the input JSON document. Unique error responses are provided when a migrated data set is requested to be recalled without waiting and for when a migrated data set is not requested to be recalled. In both cases, no job is submitted. If you have asked for a recall, without waiting, when you try the submit again, you should do so without adding the "recall" property to the JSON document or by changing the "recall" property to the value "no."

Required authorizations

See [“Required authorizations” on page 904](#).

In addition, your user ID must be authorized to run jobs on the system and be able to access any protected resources that the job might require. For information about the security considerations for job submission, see [Controlling job submission in z/OS JES2 Initialization and Tuning Guide](#) or [Authorizing the Use of Input Sources in z/OS JES3 Initialization and Tuning Guide](#).

Usage considerations

See “Usage considerations for the z/OSMF REST services” on page 4.

In addition, observe the following considerations for this request:

- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB
```

Where *JESB* is the name of the secondary JES subsystem in these examples.

Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with a JSON job document. For the contents, see “Job document” on page 936.

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See “Error report document” on page 942.

Example of submitting a job from a data set or UNIX file

Table 484 on page 922 shows variations of a PUT request that submits the job TESTJOBX to run on z/OS. In each variation, the PUT request contains a JSON statement that identifies the location of the job to be submitted.

Table 484. Variations of a PUT request for submitting a job from a data set or UNIX file.	
Location of the job	Example
Partitioned data set (fully qualified)	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A { "file" : "'MYJOBS.TEST.CNTL(TESTJOBX)'" }</pre>
Partitioned data set (non-fully qualified)	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A { "file" : "/TEST.CNTL(TESTJOBX)" }</pre>
Sequential data set	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A { "file" : "'MYJOBS.TEST.JOB1'" }</pre>
UNIX file	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A X-IBM-Intrdr-Recfm: V X-IBM-Intrdr-Lrecl: 255 X-IBM-Intrdr-Mode: TEXT { "file" : "/u/myjobs/job1" }</pre>

Example of a request that contains the job JCL

The following request submits the job TESTJOBX to run on z/OS. Here, the JCL for the job to be submitted is contained in the PUT request.

```
PUT /zosmf/restjobs/jobs HTTP/1.1
Host: zosmf1.yourco.com
Content-Type: text/plain
X-IBM-Intrdr-Class: A
X-IBM-Intrdr-Recfm: F
X-IBM-Intrdr-Lrecl: 80
X-IBM-Intrdr-Mode: TEXT

//TESTJOBX JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14
```

A sample response is shown in Figure 433 on page 923.

```
HTTP/1.1 201 Created
Date: Fri, 17 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00025", "jobname": "TESTJOBX", "subsystem": null, "owner": "IBMUSER",
  "status": "INPUT", "type": "JOB", "class": "A", "retcode": null,
  "url": "https://\\host:port\\zosmf\\restjobs\\jobs\\TESTJOBX\\JOB00025",
  "files-url": "https://\\host:port\\zosmf\\restjobs\\jobs\\TESTJOBX\\JOB00025\\files"
}
```

Figure 433. Example: Returned results of a job submission

Hold a job

For a job that has been submitted to run on z/OS, but not yet selected for processing, you can use this operation to hold the job. When held, a job is not be eligible for selection.

You can use a similar method to release the job and make is available for selection; see [“Release a job”](#) on page 926.

HTTP method and URI path

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job to be held. **<jobname>** can contain a question mark (?) that represents a single character.
 - **<correlator>** identifies the job to be held. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

"request": "hold"

Indicates a request to hold a job.

"version": "n.n"

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 903](#).

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, observe the following consideration for this request:

- This request can be directed to a secondary JES subsystem. To do so, use one of the following request formats:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid  
https://host:port/zosmf/restjobs/jobs/-JESB/correlator
```

where:

- *JESB* is the name of the secondary JES subsystem.
- The job to be held is identified by either the job name and job ID (*jobname/jobid*) or the job correlator (*correlator*).

Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 905](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
 - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 940](#).
 - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

The following request specifies that the job TESTJOBW, job ID JOB00023, is to be held. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1  
Host: zosmf1.yourco.com  
Content-Length: 40  
Content-Type: application/json
```

```
{  
  "request": "hold",  
  "version": "2.0"  
}
```

Example response

A sample response is shown in [Figure 434 on page 926](#).

```

HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1....CC20F378.....",
  "status": "0"
}

```

Figure 434. Example: Returned results of a job hold request

Release a job

For a job that has been held from execution on z/OS, you can use this operation to release the job. When released, a job is made eligible for selection to execute.

HTTP method and URI path

```

PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>

```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job to be released. **<jobname>** can contain a question mark (?) that represents a single character.
 - **<correlator>** identifies the job to be released. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

"request": "release"

Indicates a request to release a job.

"version": "n.n"

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 903](#).

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, observe the following consideration for this request:

- This request can be directed to a secondary JES subsystem. To do so, use one of the following request formats:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid  
https://host:port/zosmf/restjobs/jobs/-JESB/correlator
```

where:

- *JESB* is the name of the secondary JES subsystem.
- The job to be released is identified by either the job name and job ID (*jobname/jobid*) or the job correlator (*correlator*).

Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 905](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:

- Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 940](#).
- Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

The following request specifies that the job TESTJOBW, job ID JOB00023, is to be released. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "request": "release"
  "version": "2.0"
}
```

Example response

A sample response is shown in [Figure 435 on page 928](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1.....CC20F378.....",
  "status": "0"
}
```

Figure 435. Example: Returned results of a job release request

Change the job class

You can use this operation to change the class of a job on z/OS.

HTTP method and URI path

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:

- **<jobname>/<jobid>** identifies the job for which the class is to be changed. **<jobname>** can contain a question mark (?) that represents a single character.
- **<correlator>** identifies the job for which the class is to be changed. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

"class": "<new_job_class>"

Indicates a request to change the job class to the value **<new_job_class>**.

"version": "n.n"

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 903](#).

Required authorizations

See [“Required authorizations” on page 904](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

In addition, observe the following considerations for this request:

- The specified job class is not validated on input. To verify the success of this request, your program can issue a GET request for the job status, and check the class value in the returned JSON Job document. See [“Obtain the status of a job”](#) on page 905.
- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

where *JESB* is the name of the secondary JES subsystem.

- A request to a secondary JES subsystem must include the job name and job id, rather than a job correlator.

Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job”](#) on page 905.
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
 - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document”](#) on page 940.
 - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document”](#) on page 942.

Example request

The following request specifies job class A for the job TESTJOBW, job ID JOB00023. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "class": "A",
  "version": "2.0"
}
```

Example response

A sample response is shown in [Figure 436](#) on page 931.


```

HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1....CC20F378.....",
  "status": "0"
}

```

Figure 436. Example: Returned results of a job class change

Cancel a job

You can use this operation to cancel a job on z/OS.

HTTP method and URI path

```

PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>

```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job to be canceled. **<jobname>** can contain a question mark (?) that represents a single character.
 - **<correlator>** identifies the job to be canceled. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

Custom headers

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

"request": "cancel"

Indicates a request to cancel a job.

"version": "n.n"

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 903](#).

Required authorizations

See [“Required authorizations” on page 904](#).

In addition, your user ID must be authorized to cancel the job on the system. For information about the security considerations for job cancellation, see [Controlling job modification and cancellation in z/OS JES2 Initialization and Tuning Guide](#) or [Controlling Who Can Cancel Jobs by Job Name in z/OS JES3 Initialization and Tuning Guide](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

In addition, note that this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

where *JESB* is the name of the secondary JES subsystem in these examples. A request to a secondary JES subsystem must include the job name and job id, rather than a job correlator.

Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 905](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
 - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the

operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 940](#).

- Status code of *4nn* or *5nn* indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 942](#).

Example request

The following request cancels the job TESTJOB2, job ID JOB00084 on z/OS. To request synchronous processing by the target JES subsystem, the request includes the specification "version": "2.0".

```
PUT /zosmf/restjobs/jobs/TESTJOB2/JOB00084 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "request": "cancel",
  "version": "2.0"
}
```

Example response

A sample response is shown in [Figure 437 on page 933](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00084",
  "jobname": "TESTJOB2",
  "original-jobid": "JOB00084",
  "owner": "IBMUUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000084SY1.....CC20F378.....",
  "status": "0"
}
```

Figure 437. Example: Returned results of a job cancellation

Cancel a job and purge its output

You can use this operation to cancel a job and purge its output.

HTTP method and URI path

```
DELETE /zosmf/restjobs/jobs/<jobname>/<jobid>
DELETE /zosmf/restjobs/jobs/<correlator>
```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
 - **<jobname>/<jobid>** identifies the job to be canceled and purged. **<jobname>** can contain a question mark (?) that represents a single character.

- **<correlator>** identifies the job to be canceled and purged. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

Custom headers

You can include the following optional custom HTTP header with this request:

X-IBM-Job-Modify-Version

Use this header to specify whether the request is to be processed asynchronously or synchronously, as follows:

1.0

Specifies that the request is to be processed asynchronously. In response, the caller receives an HTTP status code of 202 Accepted, with no indication of the success or failure of the request. To verify that the initial request was performed, the caller can issue the service that is described in [“Obtain the status of a job” on page 905](#).

2.0

Specifies that the request is to be processed synchronously. In response, the caller receives an HTTP status code, which indicates the results of the request. For a successful request, the caller also receives the JSON job feedback document, which includes details about the job that was cancelled.

If this header is omitted, the request is processed asynchronously by default.

Synchronous processing is supported for JES2 only. On systems running JES3, the z/OS jobs REST interface services must run asynchronously.

For an example of how this header is specified, see [“Example request” on page 935](#).

X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 432](#) was issued in a previous request.

X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

Query parameters

None.

Required authorizations

See [“Required authorizations” on page 904](#).

In addition, your user ID must be authorized to cancel the job on the system, which allows the user to delete the job SYSOUT data sets. For security considerations for job cancellation, see [Controlling job modification and cancellation in z/OS JES2 Initialization and Tuning Guide](#) or [Controlling Who Can Cancel Jobs by Job Name in z/OS JES3 Initialization and Tuning Guide](#).

Usage considerations

See “Usage considerations for the z/OSMF REST services” on page 4.

In addition, this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service that is described in “Obtain the status of a job” on page 905.
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
 - Status code 200 indicates that the synchronous request was processed successfully. However, this status does not mean that the operation was successful. To determine the success of the operation, check the “status” property in the JSON job feedback document for a value of 0 (zero). See “Job feedback document” on page 940.
 - Status code of 4nn or 5nn indicates that an HTTP error occurred.

For HTTP errors, z/OS jobs REST interface returns error information as a JSON error report document. See “Error report document” on page 942.

Example request

The following request cancels the job TESTJOBW, job ID JOB00085 and purges its output on the z/OS system. With the inclusion of the **X-IBM-Job-Modify-Version** header set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
DELETE /zosmf/restjobs/jobs/TESTJOBW/JOB00085 HTTP/1.1
X-IBM-Job-Modify-Version: 2.0
```

Example response

A sample response is shown in Figure 438 on page 936. Because the request was processed synchronously by the target JES subsystem, the response body includes the job feedback document with details about the job that was cancelled.

```

HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB000085",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB000085",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000085SY1....CC20F380.....",
  "status": "0"
}

```

Figure 438. Example: Results of a job delete request

JSON document specifications for z/OS jobs REST interface requests

This section describes the contents of the JSON documents that are used with z/OS jobs REST interface requests.

The following JSON documents are described:

- “Job document” on page 936
- “Job notification document” on page 938
- “Job feedback document” on page 940
- “Job file document” on page 940
- “Job step data document” on page 941
- “Error report document” on page 942.

Job document

Table 485 on page 936 shows the contents of the JSON job document.

Table 485. Contents of the JSON job document	
Property	Description
jobid	Job ID.
jobname	Job name.
subsystem	The primary or secondary JES subsystem. If this value is null, the job was processed by the primary subsystem.
owner	The z/OS user ID associated with the job.
status={ INPUT ACTIVE OUTPUT }	<p>Job status. One of the following values:</p> <p>INPUT Job is in input processing.</p> <p>ACTIVE Job is running.</p> <p>OUTPUT Job is on the hardcopy output queue.</p> <p>If this value is null, the job status cannot be determined.</p>

Table 485. Contents of the JSON job document (continued)

Property	Description
type ={ JOB STC TSU }	Job type. One of the following values: JOB Batch job. STC Started task. TSU TSO/E user.
class	Job execution class.
retcode ={ ABENDUnnnn ABEND Sxxx CANCELED CC nnnn CONV ABEND CONV ERROR JCL ERROR SEC ERROR SYS FAIL }	Job completion code. One of the following values: ABENDUnnnn Job ended with the user abend code <i>nnnn</i> . ABEND Sxxx Job ended with the system abend code <i>xxx</i> . CANCELED Job was canceled. CC nnnn Job ended with the completion code <i>nnnn</i> . CONV ABEND Converter ended abnormally when processing the job. CONV ERROR Converter error when processing the job. JCL ERROR Job encountered a JCL error. SEC ERROR Job failed a security check. SYS FAIL System failure. If this value is null, the job was not yet completed.
url	Resource URL based on original HTTP request.
files-url	Resource URL for listing the spool files for the job.
job-correlator	Job correlator. If this value is null, the job was submitted to JES3.
phase	Job phase. Provides a numeric indicator of the current state of the job.
phase-name	Job phase name. Provides a text description of the specific phase of the job.
step-data	Step data information. Provides information about each step in the job, such as the step name, step number, and completion code. For more information, see “Job step data document” on page 941.
exec-system	System name of the z/OS system on which the job ran (up to 8 characters).
exec-member	Member name of the z/OS system on which the job ran (up to 8 characters).
exec-submitted	Time when the job was submitted to run (the input end time).
exec-started	Time when job execution started.
exec-ended	Time when job execution ended.

Table 485. Contents of the JSON job document (continued)

Property	Description
reason-not-running	Text identifying one or more reasons why the job is not running.

Job notification document

Table 486 on page 938 shows the contents of the JSON job notification document.

Table 486. Contents of the JSON job notification document

Property	Description
job-correlator	Job correlator. If this value is null, the job was submitted to JES3.
jobid	Job ID.
jobname	Job name.
owner	The z/OS user ID associated with the job.
class	Job execution class.
event	ready, active, or complete, based on the event that drove the HTTP POST to occur.
member	<ol style="list-style-type: none"> 1. For the ready event, indicates where the job converted and was placed onto the execution queue. 2. For the active event, indicates where the job was selected to run. 3. For the complete event, indicates where the job reached a "completed" state. Generally, it is the same as the member where the job ran, except in some error cases where that might not be true.
event-time	Timestamp indicating the date and time when the job notification event occurred. Timestamps are presented in the JSON UTC format: "yyyy-mm-ddThh:mm:ss.mmmmmmmZ".

Table 486. Contents of the JSON job notification document (continued)

Property	Description
retcode={ ABENDUnnnn ABEND Sxxx CANCELED CC nnnn CONV ABEND CONV ERROR JCL ERROR SEC ERROR SYS FAIL }	<p>Job completion code. One of the following values: For ready and active event types, property value returns as null. For event type complete, this property stands for Job completion code. One of the following values:</p> <p>ABENDUnnnn Job ended with the user abend code <i>nnnn</i>.</p> <p>ABEND Sxxx Job ended with the system abend code <i>xxx</i>.</p> <p>CANCELED Job was canceled.</p> <p>CC nnnn Job ended with the completion code <i>nnnn</i>.</p> <p>CONV ABEND Converter ended abnormally when processing the job.</p> <p>CONV ERROR Converter error when processing the job.</p> <p>JCL ERROR Job encountered a JCL error.</p> <p>SEC ERROR Job failed a security check.</p> <p>SYS FAIL System failure.</p>
completion-type	<p>Specific completion type:</p> <p>0 No completion information was received.</p> <p>1 Job ended normally.</p> <p>2 Job ended with a completion code.</p> <p>3 Job encountered a JCL error.</p> <p>4 Job was canceled.</p> <p>5 Job abended.</p> <p>6 Converter error when processing the job.</p> <p>7 Job encountered a security error.</p> <p>8 Job failed in EOM.</p> <p>9 Job failed a security check.</p> <p>10 System failure.</p>

Table 486. Contents of the JSON job notification document (continued)

Property	Description
completion-code	Completion code. Set for completion-type values 1 and 2. Otherwise, null.
abend-code	Job completed with abend code. Set for completion-type values 5 and 8. Otherwise, null. When set, one of the following values: Unnnn Job ended with the user abend code <i>nnnn</i> . Sxxx Job ended with the system abend code <i>xxx</i> .

Job feedback document

Table 487 on page 940 shows the contents of the JSON job feedback document.

Table 487. Contents of the JSON job feedback document

Property	Description
jobid	Job ID.
jobname	Job name.
original-jobid	Original job ID. If the job was processed on another system, this value represents the original job identifier that was assigned when the job was submitted on the host system. If the target system cannot assign the original job identifier, the target system assigns a new ID to the job, which is indicated as "jobid" in this document.
owner	z/OS user ID associated with the job.
member	JES2 multi-access spool (MAS) member name.
sysname	z/OS system name.
job-correlator	Job correlator. If this value is null, the job was submitted to JES3.
status={n}	job processing status. If set to zero (0), the request was processed successfully. Otherwise, there was an error. See the message property for a description of the error.
internal-code	If job processing status indicates an error (a value other than 0), this property contains the internal service routine return code. Otherwise, this property is omitted.
message	If job processing status indicates an error (a value other than 0), this property contains a description of the error. Otherwise, this property is omitted.

Job file document

Table 488 on page 940 shows the contents of the JSON job file document.

Table 488. Contents of the JSON job file document

Property	Description
jobname	Job name.

Table 488. Contents of the JSON job file document (continued)

Property	Description
recfm	<p>Record format of the file. The first character of the returned string is one of the following:</p> <p>F Fixed length records</p> <p>V Variable length records</p> <p>U Undefined length records.</p> <p>One or more subsequent characters might also be present in the returned string (in this order):</p> <p>B File has blocked records.</p> <p>S File has standard records (if fixed length format) or spanned records (if variable length format).</p> <p>M File has machine print-control characters.</p> <p>A File has ASA (ANSI) print-control characters.</p> <p>Generally, the B (blocked) and S (standard or spanned) characters are not present for JES spool files. Also, the M (machine) and A (ASA) characters are mutually exclusive.</p>
byte-count	Number of bytes on spool consumed by the spool file. The value can be zero (0). This field is integer data type.
record-count	Number of records in the spool file. The value can be zero (0). This field is integer data type.
job-correlator	Job correlator. If this value is null, the job was submitted to JES3.
class	Class that is assigned to the spool file.
jobid	Job ID.
id	Data set number (key). This field is integer data type.
ddname	DDNAME for the data set creation.
records-url	Resource URL for retrieving the spool file contents for the job.
lrecl	Specifies the length, in bytes, for fixed-length records and the maximum length for variable-length records.
subsystem	The primary or secondary JES subsystem. If the value is null, the job was processed by the primary subsystem.
stepname	Step name for the step that created this data set. The value can be null.
procstep	Procedure name for the step that created this data set. The value can be null.

Job step data document

Table 489 on page 942 shows the contents of the JSON job step data document.

Table 489. Contents of the JSON job step data document

Property	Description
active	Value is set to <i>true</i> if the step is running. Otherwise, the value is <i>false</i> .
smfid	The SMF ID of the system where the step is running.
step-number	Step number.
selected-time	Date and time the step started. Not returned if the step is still active.
owner	The z/OS user ID associated with the job.
program-name	Name of the program to be run by the job step. This value is retrieved from the EXEC statement.
step-name	Name specified for the step on the EXEC statement.
path-name	Path to the program in the z/OS UNIX System Services (z/OS UNIX) file system that is run by the job step. Not returned if the step is active.
substep-number	Step number in the sequence of steps that run z/OS UNIX programs. Not returned if the step is still active.
end-time	Date and time the step completed. Not returned if the step is still active.
proc-step-name	Name of the procedure to be run by the job step. This value is retrieved from the EXEC statement.
completion={ ABENDUnnnn ABEND Sxxx CANCELED CC nnnn FLUSHED }	<p>Step completion code. One of the following values:</p> <p>ABENDUnnnn Step ended with the user abend code <i>nnnn</i>.</p> <p>ABEND Sxxx Step ended with the system abend code <i>xxx</i>.</p> <p>CANCELED Step was canceled.</p> <p>CC nnnn Step ended with the completion code <i>nnnn</i>.</p> <p>FLUSHED Step was not processed.</p> <p>Not returned if the step is active.</p>
abend-reason-code	Abend reason code. Returned if the step is completed and an abend occurs. Not returned if the step is active.

Error report document

Table 490 on page 942 shows the contents of the JSON error report document.

Table 490. Contents of the JSON error report document

Property	Description
category	Error category. This field is integer data type.
rc	Return code. This field is integer data type.
reason	Reason code. This field is integer data type.
message	Message that describes the error.

Table 490. Contents of the JSON error report document (continued)	
Property	Description
details	(Optional) Array of strings containing additional message details.

For the meanings of the category, return code, and reason code fields, see [“Error reporting categories”](#) on page 943.

Error reporting categories

This section describes the error categories and associated error codes that can be returned in the JSON error report document, described in [“Error report document”](#) on page 942.

Categories

Table 491 on page 943 shows the error categories that are defined for errors that are returned in z/OS jobs REST interface operations.

Table 491. Error categories for z/OS jobs REST interface operations			
Category	Ordinal Value	Description	Where the error details are described
Dynalloc	1	Dynamic allocation errors.	“Category 1 – Dynamic allocation error” on page 943
VSAM API	3	Errors that are produced or detected by the Java/ JNI/ C/ HLASM/ VSAM layer.	“Category 3 – VSAM API error” on page 945
VSAM system	4	Errors that are produced or detected by VSAM. The return code and reason code are VSAM-specific.	“Category 4 – VSAM system error” on page 945
VSAM ABEND	5	ABEND information that results from VSAM failures.	“Category 5 – VSAM ABEND error” on page 945
Service	6	Errors that are produced or detected in the service layer.	“Category 6 – Service error” on page 945
Unexpected	7	Unexpected errors detected.	“Category 7 – Unexpected error” on page 951
SSI extended status	8	Errors that are produced or detected by the extended status function call of the subsystem interface (SSI Function Code 80).	“Category 8 – SSI extended status error” on page 951
CIM	9	Errors that are produced or detected by the Common Information Model (CIM) interface.	“Category 9 – Common Information Model (CIM) error” on page 952
SSI job modify	10	Errors that are produced or detected by the job modify function call of the subsystem interface (SSI Function Code 85).	“Category 10 – SSI job modify error” on page 952

Category 1 – Dynamic allocation error

Table 492 on page 944 shows the possible conditions for this error category.

Table 492. Category 1 errors

Return code (rc)	Reason	Message	Description
n	0	Error allocating internal reader, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate the internal reader for job submission. In the message, <i>RC</i> is error data from the dynamic allocation request (SVC 99).</p> <p>To diagnose the error, convert the <i>RC</i> value from decimal to a 4-byte hexadecimal value, which provides the dynamic allocation error code and information code, as follows:</p> <ul style="list-style-type: none"> High-order two bytes indicate the error code from the dynamic allocation request (field S99ERROR in the input request block S99RB). Low order two bytes indicate the information code from the dynamic allocation request (field S99INFO in the input request block S99RB). <p>For information about dynamic allocation and its related error and information codes, see Dynamic allocation in z/OS MVS Programming: Authorized Assembler Services Guide.</p>
n	1	Error allocating input data set: %s, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate a ddname for the data set specified as the source for the input job. In the message, <i>RC</i> is the return code from the BPXWDYN service.</p> <p>For information about BPXWDYN and its related return codes, see BPXWDYN: A text interface to dynamic allocation and dynamic output in z/OS Using REXX and z/OS UNIX System Services.</p>
n	2	Error allocating spool file: job '%s' spool file id %d, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate the requested spool file. Perhaps, a thread is attempting to allocate the spool file while another thread is requesting to cancel the job and purge its output.</p> <p>In the message, <i>RC</i> is error data from the dynamic allocation request (SVC 99). Both decimal and hexadecimal values are provided in the message.</p> <p>The hexadecimal value provides the dynamic allocation error code and information code, as follows:</p> <ul style="list-style-type: none"> High-order two bytes indicate the error code from the dynamic allocation request (field S99ERROR in the input request block S99RB). Low order two bytes indicate the information code from the dynamic allocation request (field S99INFO in the input request block S99RB). <p>For information about dynamic allocation and the meanings of the error code and information code, see Dynamic allocation in z/OS MVS Programming: Authorized Assembler Services Guide.</p>

Category 3 – VSAM API error

Table 493 on page 945 shows the possible conditions for this error category.

Table 493. Category 3 errors			
Return code (rc)	Reason	Message	Description
4	1	Incorrect JesVsam handle	
4	2	VSAM file is not open	
4	3	Record length %d > lrecl %d	Writing a record to a VSAM file failed because an incorrect record length was specified.
4	4	Could not write JCL to internal reader	An I/O exception occurred when JCL to the internal reader.
8	0	JesVsam get failed	Buffer too small to hold the VSAM record.
255	0	JesVsam native buffer malloc failed	

Category 4 – VSAM system error

Table 494 on page 945 shows the possible conditions for this error category.

Table 494. Category 4 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	For descriptions of the specific return and reason codes, see the VSAM publications.

Category 5 – VSAM ABEND error

Table 495 on page 945 shows the possible conditions for this error category.

Table 495. Category 5 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The values <i>n</i> and <i>m</i> indicate the ABEND return code and reason code.

Category 6 – Service error

Table 496 on page 946 shows the possible conditions for this error category.

Table 496. Category 6 errors

Return code (rc)	Reason	Message	Description
4	1	Incorrect Internal Reader mode: %s. Must be one of TEXT RECORD BINARY	Request header X-IBM-Intrdr-Mode specified a value that is not valid. Valid values are TEXT, BINARY, or RECORD.
4	2	Incorrect Internal Reader parameters: %s. Fixed records are required for binary mode	The internal reader characteristics form a combination that is not valid. If you specify the value BINARY for the X-IBM-Intrdr-Mode request header, you must specify the value F for the X-IBM-Intrdr-Recfm request header.
4	3	Request does not contain '%s' content	Job modification requests must have a content type of application/json.
4	4	Value of %s query parameter is not valid	The query parameter that was identified in the message either contains incorrect characters or exceeds the allowable length. In the message, the query parameter is <i>owner</i> , <i>prefix</i> , <i>jobid</i> , or <i>job-correlator</i> .
4	5	Update request is not 'cancel'	In the job modification request, the "request" property is set to an incorrect value. The required value is "cancel."
4	6	Request does not contain a valid job update request	In the job modification request, the input document does not specify a valid property. The valid properties are: <ul style="list-style-type: none"> • "request" • "class"
4	7	No match for method %s and pathInfo='%s'	The supplied servlet path information (pathinfo) does not match any expected string for the HTTP method that was specified.
4	8	POST requests not supported	For standard REST requests, the POST HTTP method is not allowed. To avoid this message, include the X-IBM-Requested-Method header to send the request through the POST verb.
4	9	Job submission error. Record length %d too long for JCL submission, maxlen=%d	The check for record mode job submission failed.
4	10	No job found for reference: '%s'	The job modification request specifies a job that does not exist.

Table 496. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	11	Record range '%s' is not valid for spool file record request	Request header X-IBM-Record-Range specified a value that is not valid. The content range must be specified by using one the following formats: SSS-EEE Where <i>SSS</i> identifies the start record and <i>EEE</i> identifies the end record to be retrieved. Both values are relative offsets (0-based). When <i>EEE</i> is set to 0, records through the end of the spool file are retrieved. SSS,NNN Where <i>SSS</i> identifies the start record and <i>NNN</i> identifies the number of records to be retrieved.
4	12	Job '%s' does not contain spool file id %d	
4	13	Job input was not recognized by system as a job	The job was submitted without a job statement or with unrecognized (non-JCL) content.
4	14	Unsupported encoding: %s	On a job submission request, the Content-Type request header specified an unsupported character set (charset).
4	15	DD names are not supported for submit input	The filename property in the JSON document started with //DD:, indicating the dd:ddname syntax. This syntax is not supported.
4	16	Data set not found	The data set specified in the JSON document that was provided on the submit job interface was not found. Possibly, the data set is not cataloged.
4	17	Submit input data does not start with a slash	This error occurs when the first character of the input job is not the EBCDIC slash character. Possible causes include: <ul style="list-style-type: none"> The Content-Type request header is set to text/plain when a JSON document that names the source of the input job is also used. The input data set or file does not contain EBCDIC data.
4	18	Submit input filename must be absolute path: %s	The z/OS UNIX file specification in the JSON document was not an absolute path.
4	19	Internal reader mode must be RECORD for data set submission: %s	If specified, the internal reader mode must be set to RECORD when a job is submitted from a data set.
4	20	Service not implemented: %s	The requested service is not supported. The variable text %s contains additional information.

Table 496. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	22	Internal reader RECFM (%s) does not match data set RECFM (%s): %s	If specified, the internal reader record format must match the record format of the existing data set when a job is submitted from a data set.
4	23	Internal reader LRECL (%d) does not match data set LRECL (%d): %s	If specified, the internal reader logical record length must match the logical record length of the existing data set when a job is submitted from a data set.
4	24	Content type '%s' not valid for internal reader mode '%s'	The values that are specified for Content-Type and internal reader mode are not a supported combination.
4	25	JCL symbol name '%s' is not valid	The specified symbol name does not match the syntax rules for a JCL symbol or start with the characters 'SYS'.
4	26	JCL symbol '%s' value exceeds maximum length	The value that is supplied for the specified symbol name exceeds the maximum value length of 255 characters.
4	27	No value supplied for JCL symbol '%s'	For a JCL symbol to be defined, it must have a non-null, non-blank value.
4	28	Maximum number of JCL symbols exceeded	An attempt was made to define more than 128 symbols, which is not allowed.
4	29	User correlator '%s' is not valid	The specified user correlator (X-IBM-User-Correlator) does not match the syntax rules for a user correlator.
4	30	Notification URL '%s' exceeds maximum length	The specified notification URL exceeds the maximum value of 2083 characters.
4	31	Request header not supported by primary JES subsystem: %s	The request header that is identified in the message is not supported by the primary job entry subsystem.
4	32	Error parsing JSON input	<p>On a job submission request, the header Content-Type specified application/json. However, an exception occurred when the input JSON document was processed.</p> <p>Possible causes include:</p> <ul style="list-style-type: none"> • JCL stream was provided instead of a JSON document. • JSON document was malformed. • Required "file" property was not provided. • Value for the "recall" property was not valid. <p>For details about the exception, check the z/OSMF logs.</p>

Table 496. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	33	Data set is migrated: %s	The JSON document specifies a z/OS data set that was migrated. No recall was issued. No job was submitted.
4	34	Recall issued for migrated data set: %s	The JSON document specifies a z/OS data set that was migrated. A recall without waiting was issued. No job was submitted.
4	35	Error recalling data set, RC=%d	An error occurred during the recall of a migrated data set. The return code from the ARCHRCAL service is included in the message. For information about the ARCHRCAL service and return codes from the DFSMSHsm user macros, see ARCHRCAL: Recalling a data set in z/OS DFSMSHsm Managing Your Own Data .
4	36	Incorrect internal reader class: %s. Must be one character in length.	The internal reader class request header specified a value that is not valid. The class must be one character in length.
4	37	Incorrect job update version requested: %s.	In the job modification request, the "version" property or the X-IBM-Job-Modify-Version request header is set to an incorrect value. The valid values are "1.0" or "2.0".
4	38	The query parameters <i>search</i> and <i>research</i> cannot be used together.	The search and research query parameters are mutually exclusive; they cannot be used together. Specify either search or research, but not both.
4	39	The query parameters <i>search</i> and <i>research</i> are allowed only on <i>mode=text</i> requests.	The query parameters search and research cannot be used when mode is set to a value other than text, such as mode=record or mode=binary.
4	40	The query parameter <i>maxreturnsize</i> is not a valid integer.	The query parameter maxreturnsize must be set to a valid integer value, for example,maxreturnsize=100.
4	41	The value of X-IBM-Notification-Options is too long. Supply a value that contains no more than 4096 characters.	The specified events exceed the maximum value of 4096 characters.

Table 496. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	42	The X-IBM-Notification-Options header is not valid. Supply a valid value for the header.	The value of X-IBM-Notification-Options is not a JSON String or the value is not valid.
8	1	Unable to query information about submitted job: %s	The job status for the submitted job was not obtained within the timeout period (3 seconds).
8	2	EOF encountered before all requested bytes read (%d / %d)	Internal read state error. The expected number of bytes were not available to be read before the end of file (EOF) was reached.
8	3	Range start is beyond end of spool file %d for job %s	In a GET request for a range of records for a spool file, the X-IBM-Record-Range header specified a record start value that is beyond the end of the spool file.
8	4	Cannot advance spool file more than Integer.MAX_VALUE. DD= %s	In a GET request for a range of records in a spool file, the X-IBM-Record-Range header specified a record start value that was greater than 2**31-1.
8	5	Error opening input data set: %s	An error occurred opening the input z/OS data set. For a message with additional information, check the z/OSMF logs.
8	6	Error reading submit input data	An error occurred reading the submit input data. For a message with additional information, check the z/OSMF logs.
8	7	Error opening input file: %s	An error occurred opening the input z/OS UNIX file. For a message with additional information, check the z/OSMF logs.
8	8	IAZSYMBL error defining %s	The JES symbol definition service (IAZSYMBL) failed during the attempt to define the specified information. In the message, %s is one of the following values: <ul style="list-style-type: none"> • User correlator • Notification URL • One or more JCL symbols. For details about the IAZSYMBL error, check the z/OSMF logs.
12	1	Not authorized to access spool file	An authorization check failed during the attempt to open the requested spool file.

Table 496. Category 6 errors (continued)			
Return code (rc)	Reason	Message	Description
12	2	Not authorized to submit job	An authorization check failed during the attempt to open the internal reader to submit a job.
12	3	User not authorized to issue a CIM request	<p>CIM detected an authentication or authorization failure during the request. To use the requested service, the user must be authorized to use the CIM server and be permitted to the JES2-JES3Jobs CIM provider.</p> <p>The requested service was one of the following:</p> <ul style="list-style-type: none"> • Hold a job. • Release a job. • Change the job class. • Cancel a job. • Delete a job. <p>CIM provides jobs (CFZSEC and CFZRCUST) to help you configure the CIM server, including security authorizations and file system customization. For more information, see Quick guide: CIM server setup and verification in z/OS Common Information Model User's Guide.</p>

Category 7 – Unexpected error

Table 497 on page 951 shows the possible conditions for this error category.

Table 497. Category 7 errors			
Return code (rc)	Reason	Message	Description
16	1	Server error occurred	For details about the exception, check the z/OSMF logs.

Category 8 – SSI extended status error

Table 498 on page 951 shows the possible conditions for this error category.

Table 498. Category 8 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The return and reason codes (<i>n,m</i>) are set from the extended status function of subsystem interface (SSI) Function Code 80 return code and the subsystem options block (SSOB) return code. The details property of the JSON error report document contains a message with more information. See “Error report document” on page 942.

Category 9 – Common Information Model (CIM) error

Table 499 on page 952 shows the possible conditions for this error category.

Table 499. Category 9 errors			
Return code (rc)	Reason	Message	Description
4	2	Incorrect jobname: "%s"	Before the CIM service call, the job name was found to be null or an empty string.
4	3	Incorrect jobid: "%s"	Before the CIM service call, the job ID was found to be null or an empty string.
4	4	Incorrect JES type	Before the CIM service call, an incorrect JES type (not JES2 or JES3) was detected.
4	5	Incorrect job class: "%s"	Before the CIM service call, the job class was found to be null or an empty string.
8	—	Varies	CIM internal error. An error occurred during setup or invocation of the CIM service.
12	<i>m</i>	Error returned from CIM job {Cancel Hold Release Request Property Change}service	CIM response error. Reason (<i>m</i>) is the reason code that was returned from CIM. The "details" property of the JSON error report document contains the CIM response text, if any. See “Error report document” on page 942 .
16	—	CIM connection failure	<p>A connection exception was encountered when the request was processed. This error can occur during periods of concurrent high usage of the REST interfaces. Usually, the reason for the failure is a connection refused due to overload of the server. The application can try the request again. The number of retry attempts that are needed depends on how much work is being requested of the server.</p> <p>One of the following services was requested:</p> <ul style="list-style-type: none">• Hold a job.• Release a job.• Change the job class.• Cancel a job.• Delete a job.

Category 10 – SSI job modify error

Table 500 on page 953 shows the possible conditions for this error category.

Table 500. Category 10 errors

Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The return and reason codes (<i>n,m</i>) are set from the job modify function call of the subsystem interface (SSI Function Code 85) return code and the subsystem options block (SSOB) return code. The "details" property of the JSON error report document contains a message with more information. See "Error report document" on page 942.

z/OS Management Services Catalog services

The z/OS Management Services Catalog (zMSC) task improves how z/OS system programmers manage their z/OS environment.

The z/OS Management Services Catalog REST APIs

Table 501 on page 953 lists the operations that the z/OS Management Services Catalog task provides.

Table 501. z/OS Management Services Catalog services: operations summary

Operation name	HTTP method and URI path
"List the services in the catalog" on page 956	GET /zosmf/mgmt-services/rest/catalog-services
"Get the details of a service from the catalog" on page 962	GET /zosmf/mgmt-services/rest/catalog-services/{object-id}
"Get the list of categories" on page 975	GET /zosmf/mgmt-services/rest/categories
"Get category details" on page 978	GET /zosmf/mgmt-services/rest/categories/{object-id}
"List the service submissions" on page 979	GET /zosmf/mgmt-services/rest/service-instances
"Get the details of a service submission" on page 990	GET /zosmf/mgmt-services/rest/service-instances/{object-id}
"Create a new service submission" on page 982	POST /zosmf/mgmt-services/rest/service-instances
"Delete a service submission" on page 989	DELETE /zosmf/mgmt-services/rest/service-instances/{object-id}
"Modify an existing service submission" on page 995	PATCH /zosmf/mgmt-services/rest/service-instances/{object-id}
"Perform an action on a service submission" on page 997	POST /zosmf/mgmt-services/rest/service-instances/{object-id}/actions/{action}

Table 501. z/OS Management Services Catalog services: operations summary (continued)

Operation name	HTTP method and URI path
“Get the list of JCL JOB statements” on page 999	GET /zosmf/mgmt-services/rest/settings/job-statements
“Get the list of target systems” on page 1001	GET /zosmf/mgmt-services/rest/settings/target-systems

Using the Swagger interface

You can use the Swagger interface to display information about the z/OS Management Services Catalog REST APIs. For more information, see [Chapter 1, “Using the z/OSMF REST services,” on page 1.](#)

User Roles

Role	Capabilities
User	<p>Users have access to the Catalog, Activity, and History pages. Users can submit services from the Catalog, manage queued and active service submissions in Activity, and access completed and terminated service submissions in History. Users can see service submissions from other users and administrators.</p> <p>A service submission that has been started but is not yet submitted is saved in My drafts and cannot be seen by other users or administrators.</p> <p>Users can control their notification preferences by using Settings.</p>
Administrator	<p>Administrators have additional authority that grants them access to the Administration page and the plug-in Global settings page.</p> <p>Administrators can manage existing services, create new services, and request approval to publish new services to the Catalog. Administrators can manage the plug-in's Global settings and take actions on service submissions that are created by other users.</p>
Publishing approver	<p>The publishing approver role authorizes a user ID to be assigned as an approver of services that are requested to be published in the Catalog. User IDs given this role must be that of a real user that can review and approve requests to publish services so that they are available in the Catalog page.</p> <p>Approvers are assigned by an administrator in the Publishing approvals table of Global Settings.</p> <p>Approvers have access to the Administration page and all services for which they are an approver.</p>

Role	Capabilities
RunAsUser step approver	<p>A user ID must have this role if a service's underlying workflow definition file requires the user ID to approve the use of a runAsUser step. User IDs given this role must be that of a real user that can review and approve the use of the runAsUser step. Do not use functional or application IDs as approvers.</p> <p>A runAsUser step is a step in the workflow that is performed by a specific user ID that might not be the user that runs the workflow. The user ID that the step is performed as is not necessarily the same as the user ID that approves the step.</p> <p>Every runAsUser step has an assigned user ID to approve it. This user ID's approval is required to publish the service in the Catalog.</p> <p>RunAsUser step approvers have access to the Administration page and all services for which they are an approver.</p>
RunAsUser user ID	<p>This role is required for any user ID assigned as the runAsUser for a runAsUser step in a workflow definition. Authorization to this role is checked when you create a new service from a workflow definition that contains runAsUser steps. It is also checked when the Workflows task runs a runAsUser step for workflow instances that are created by service submissions.</p> <p>This role does not grant any access to z/OS Management Services Catalog.</p>

Authorization requirements

Use of the z/OS Management Services Catalog API requires the client to be authenticated. For information about client authentication, see [“Authenticating to z/OSMF” on page 3](#).

Additionally, users must be authorized to one of the z/OS Management Services Catalog user roles. For more information, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request that is succeeded and resulted in the creation of an object.

HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

HTTP 204 No content

The request succeeded, but no content is available to be returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 401 Unauthorized

The submitter of the request did not authenticate to z/OSMF.

HTTP 403 Forbidden

Returned when the user is not authorized to perform the requested operation.

HTTP 404 Not found

The requested resource does not exist.

HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

HTTP 500 Internal Error

Returned when unexpected internal processing failures occur.

Error response content

For the HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object that contains the following fields:

Table 502. Response from a z/OS Management Services Catalog request failure		
Field	Type	Description
httpStatus	String	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageId	String	Message identifier for the error.
messageText	String	Message describing the error.
additionalInfo	String	Additional information that describes the error.
debug	String	Debug information about for the error.

Error logging

Errors from the z/OS Management Services Catalog are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

List the services in the catalog

This operation gets the list of services that are defined in z/OS Management Services Catalog (zMSC).

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/catalog-services
```

Query parameters

You can specify the following query parameters on this request. Services matching all query parameters are returned, except for the summary query parameter. The summary query parameter specifies whether to return summary or complete details for each service. For example, `/zosmf/mgmt-services/rest/catalog-services?categoryName=foo&summary=true&state=published` returns summary details for all the services that are in the category that is named foo and have the published state.

Table 503. Query parameters		
Query Parameter	Required/Optional	Description
serviceName	Optional	The value is a regular expression. Services that match the regular expression are returned.

Table 503. Query parameters (continued)		
Query Parameter	Required/Optional	Description
categoryName	Optional	The value is a regular expression. Services with a category name that match the regular expression are returned.
state	Optional	Services with a state that matches the specified state are returned. Valid values are: <ul style="list-style-type: none"> • published • pending-publish-approval • suspended (only applicable to published services) • approved • archived
summary	Optional	true false Specifies whether a summary or full details of each service is returned. If the summary is not specified, full details are returned. The summary properties are: <ul style="list-style-type: none"> • csName • objectId • objectUri • objectUrl • csDescription • csCategoryObjectId • csCategoryName • csState • csTargetSystems • csRunOnAllSystems • mgmtsvcsVersion • mgmtsvcsVersionSupported

Description

This operation returns the list of services that are defined in z/OS Management Services Catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request returned the list of services. A response body is provided, as described in [Table 504 on page 958](#).

Request Content

None.

Authorization Requirements

The user's z/OS user ID must have access to z/OSMF and must be defined as a z/OS Management Services Catalog user.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned, and the response body is provided, as described in Table 504 on page 958. If query parameters were specified that resulted in no results to be returned, HTTP status code 204 (No Content) is returned.

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the catalog service returns a response body that contains a JSON object with details about the services. For more information, see [Table 504 on page 958](#).

Table 504. Array of objects	
Type	Description
Array of catalog service objects. For more information, see Table 505 on page 963 .	Array of catalog service objects. The array is filtered based on any query parameters that were provided.

Example HTTP interaction

The example in [Figure 439 on page 958](#) shows a request to get the list of services in z/OS Management Services Catalog.

```
GET /zosmf/mgmt-services/rest/catalog-services
```

Figure 439. Sample request to get the list of services in z/OS Management Services Catalog.

The following example shows a response body that contains a single service.

```
[
  {
    "csName": "Great New Service",
    "csDescription": "A description of this great new services.",
    "objectId": "128c778f-94444030-b2d13509-87640705",
    "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "csCategoryObjectId": "128c778f-94444030-b2d13509-87640705",
    "csCategoryName": "Catalog Management",
    "csState": "published",
    "csCreatedTime": "2019-11-15T19:12:45.223Z",
    "csDefaultInstanceExpiryInterval": "PT98H",
    "csCreatedBy": "suser1",
    "csLastModifiedTime": "2019-12-22T09:12:48.291Z",
    "csLastModifiedBy": "suser2",
    "csOriginalWfDefinitionPath": "/u/suser1/workflows/createZosThing.xml",
    "csOriginalWfVariablePropertiesPath": "/u/suser1/workflows/wfVariable.properties",
    "mgmtsvcsVersion": 100,
    "mgmtsvcsVersionSupported": true,
    "csExpiryUpdateableInRunner": "false",
    "csTargetSystems": ["sys1", "sys2", "sys3"],
    "csRunOnAllSystems": true,
    "csChangeRecordAlwaysRequired": false,
    "csChangeRecordRequiredSystems": ["sys1", "sys2", "sys3"],
```

```

        "csApprovals": [
            {
                "csApprovalId": "e821ee1-2b4b483-926728e-fb6114f",
                "csApprovers": [
                    "ibmuser"
                ],
                "csApprovalIsRunAsUser": false,
                "csApprovalStatus": "PENDING"
            }
        ],
        "csBeingModifiedBy": "user21",
        "csPages": [
            {
                "csPageName": "Untitled",
                "csPageDescription": "Detailed information about this page",
                "csInputOrder": [
                    "remountOption",
                    "fileSystemName"
                ]
            }
        ],
        "csInputDefinitions": {
            "remountOption": {
                "csInputWfVarName": "remountOption",
                "csInputLabel": "Remount Option",
                "csInputDescription": "This variable determines the behavior of the remounting
process",
                "csInputType": "string",
                "csInputWfVariableCategory": "Configuration",
                "csInputErrorMessage": "The value entered is not valid",
                "csInputFilePathVariable": false,
                "csInputValidation": {
                    "maxLength": 100000
                },
                "csInputValues": [
                    "SWITCH",
                    "RDRW",
                    "READ",
                    "SAMEMODE"
                ],
                "csInputValuesMetadata": {
                    "SWITCH": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "RDRW": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "READ": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "SAMEMODE": {
                        "csChoiceValueDefinedByWorkflow": true
                    }
                },
                "csInputValueMustBeChoice": true,
                "csInputWidget": "radio-button",
                "csInputIncludePrefix": true,
                "csInputDefault": "SWITCH",
                "csInputRequiredAtCreate": false,
                "csInputRequired": false,
                "csInputPromptAtCreate": true
            },
            "fileSystemName": {
                "csInputWfVarName": "fileSystemName",
                "csInputLabel": "File System Name",
                "csInputDescription": "The fully qualified name of the date set containing the
file system",
                "csInputType": "string",
                "csInputWfVariableCategory": "Configuration",
                "csInputErrorMessage": "The value entered is not valid",
                "csInputFilePathVariable": false,
                "csInputValidation": {
                    "regularExpression": "[A-Z#$@]",
                    "validationType": "DSNAME"
                },
                "csInputValues": [],
                "csInputValuesMetadata": {},
                "csInputWidget": "text-box",
                "csInputIncludePrefix": true,
                "csInputRequiredAtCreate": false,
                "csInputRequired": false,
                "csInputPromptAtCreate": true
            }
        }
    },
    {

```

```

        "csInputLogic": {
            "unmountOption": {
                "csControlInputDefinitionId": "unmountOption",
                "csDependentLogics": [
                    {
                        "csControlInputDefinitionComparatorValues": [
                            "REMOUNT"
                        ],
                        "csControlInputDefinitionComparator": "equal",
                        "csDependentInputDefinitionIds": [
                            "remountOption"
                        ]
                    }
                ]
            }
        },
        "csRunAsUserSteps": [
            {
                "stepName": "Step1",
                "stepTitle": "Step 1",
                "stepDescription": "Brief description about the step",
                "stepRunAsUser": "runAsUID",
                "stepRunAsUserSubstitutable": false,
                "stepApprovals": [
                    "stepRunAsUserApprovers": ["apprId1", "apprId2"],
                    "stepRunAsUserApproverSubstitutable": false
                ]
            },
            {
                "stepName": "step2",
                "stepTitle": "Step 2",
                "stepDescription": "Brief description about the step",
                "stepRunAsUser": "runAsUID",
                "stepRunAsUserSubstitutable": false,
                "stepApprovals": [
                    "stepRunAsUserApprovers": ["apprId2"],
                    "stepRunAsUserApproverSubstitutable": false
                ]
            }
        ],
        "csActivityTimeline": [{
            "actionName": "REQUEST-PUBLISH-APPROVAL",
            "actionComment": null,
            "actionTime": "2020-07-09T20:17:56.801Z",
            "actionUser": "zosmfad",
            "actionAdditionalInformation": null
        }]
    },
    {
        "csName": "Great New New Service",
        "csDescription": "A description of this great new services.",
        "objectId": "128c778f-94444030-b2d13509-84629047",
        "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-84629047",
        "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-84629047",
        "csCategoryId": "128c778f-94444030-84629047-87640705",
        "csCategoryName": "Catalog Management",
        "csState": "published",
        "csCreatedTime": "2019-11-15T19:12:45.223Z",
        "csDefaultInstanceExpiryInterval": "PT98H",
        "csCreatedBy": "suser1",
        "csLastModifiedTime": "2019-12-22T09:12:48.291Z",
        "csLastModifiedBy": "suser2",
        "csOriginalWfDefinitionPath": "/u/suser1/workflows/createZosThing.xml",
        "csOriginalWfVariablePropertiesPath": "/u/suser1/workflows/wfVariable.properties",
        "mgmtsvcsVersion": 100,
        "mgmtsvcsVersionSupported": true,
        "csExpiryUpdateableInRunner": "false",
        "csTargetSystems": ["sys1", "sys2", "sys3"],
        "csRunOnAllSystems": true,
        "csChangeRecordAlwaysRequired": true,
        "csChangeRecordRequiredSystems": [],
        "csBeingModifiedBy": "user21",
        "csPages": [{
            "csPageName": "Untitled",
            "csPageDescription": "Detailed information about this page",
            "csInputOrder": [
                "remountOption",
                "fileSystemName"
            ]
        }]
    }
}],

```

```

    "csInputDefinitions": {
      "remountOption": {
        "csInputWfVarName": "remountOption",
        "csInputLabel": "Remount Option",
        "csInputDescription": "This variable determines the behavior of the remounting
process",
        "csInputType": "string",
        "csInputWfVariableCategory": "Configuration",
        "csInputErrorMessage": "The value entered is not valid",
        "csInputFilePathVariable": false,
        "csInputValidation": {
          "maxLength": 1000000
        },
        "csInputValues": [
          "SWITCH",
          "RDRW",
          "READ",
          "SAMEMODE"
        ],
        "csInputValuesMetadata": {
          "SWITCH": {
            csChoiceValueDefinedByWorkflow : true
          },
          "RDRW": {
            csChoiceValueDefinedByWorkflow : true
          },
          "READ": {
            csChoiceValueDefinedByWorkflow : true
          },
          "SAMEMODE": {
            csChoiceValueDefinedByWorkflow : true
          }
        },
        "csInputValueMustBeChoice": false,
        "csRestrictInputValueChoices": true,
        "csInputWidget": "radio-button",
        "csInputIncludePrefix": true,
        "csInputDefault": "SWITCH",
        "csInputRequiredAtCreate": false,
        "csInputRequired": false,
        "csInputPromptAtCreate": true
      },
      "fileSystemName": {
        "csInputWfVarName": "fileSystemName",
        "csInputLabel": "File System Name",
        "csInputDescription": "The fully qualified name of the date set containing the
file system",
        "csInputType": "string",
        "csInputWfVariableCategory": "Configuration",
        "csInputErrorMessage": "The value entered is not valid",
        "csInputFilePathVariable": false,
        "csInputValidation": {
          "regularExpression": "[A-Z#$$@]",
          "validationType": "DSNAME"
        },
        "csInputValues": [],
        "csInputValuesMetadata": {},
        "csInputWidget": "text-box",
        "csInputIncludePrefix": true,
        "csInputRequiredAtCreate": false,
        "csInputRequired": false,
        "csInputPromptAtCreate": true
      }
    },
    "csInputLogic": {
      "unmountOption": {
        "csControlInputDefinitionId": "unmountOption",
        "csDependentLogics": [
          {
            "csControlInputDefinitionComparatorValues": [
              "REMOUNT"
            ],
            "csControlInputDefinitionComparator": "equal",
            "csDependentInputDefinitionIds": [
              "remountOption"
            ]
          }
        ]
      }
    }
  },
  "csRunAsUserSteps": [
    {

```

```

        "stepName": "Step1",
        "stepTitle": "Step 1",
        "stepDescription": "Brief description about the step",
        "stepRunAsUser": "runAsUIId",
        "stepRunAsUserSubstitutable": false,
        "stepApprovals": [
            "stepRunAsUserApprovers": ["apprId1", "apprId2"],
            "stepRunAsUserApproverSubstitutable": false
        ]
    },
    {
        "stepName": "step2",
        "stepTitle": "Step 2",
        "stepDescription": "Brief description about the step",
        "stepRunAsUser": "runAsUIId",
        "stepRunAsUserSubstitutable": false,
        "stepApprovals": [
            "stepRunAsUserApprovers": ["apprId2"],
            "stepRunAsUserApproverSubstitutable": false
        ]
    }
],
"csActivityTimeline": [{
    "actionName": "REQUEST-PUBLISH-APPROVAL",
    "actionComment": null,
    "actionTime": "2020-07-09T20:17:56.801Z",
    "actionUser": "zosmfad",
    "actionAdditionalInformation": null
}]
}
]

```

Get the details of a service from the catalog

Use this operation to get the details of a specific service from the catalog in z/OS Management Services Catalog (zMSC).

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/catalog-services/{object-id}
```

In this request:

object-id

Identifies the service to be retrieved from the z/OS Management Services Catalog.

Query parameters

None.

Description

This operation retrieves the details of a specific catalog service from the catalog within the z/OS Management Services Catalog (zMSC) task.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request returned the details for the requested catalog service. A response body is provided as, described in the [“Response content”](#) on page 963 section.

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or zMSC Administrator.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned, and the response body is provided, as described in “Response content” on page 963. If query parameters were specified that resulted in no results to be returned, HTTP status code 204 (No Content) is returned.

For more information about error status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service returns a response body that contains a JSON object with details about the catalog service. For more information, see:

- [Table 505 on page 963](#)
- [Table 506 on page 966](#)
- [Table 507 on page 969](#)
- [Table 508 on page 969](#)
- [Table 509 on page 970](#)
- [Table 510 on page 970](#)
- [Table 511 on page 971](#)
- [Table 512 on page 971](#)
- [Table 513 on page 972](#)
- [Table 514 on page 973](#)
- [Table 515 on page 973](#)

Table 505. Fields for a catalog service object			
Field	Type	Guaranteed/ Optional	Description
csName	String	Guaranteed	Descriptive name for the catalog service. The name must be unique within its assigned category. The name cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
csDescription	String	Optional	Description of the catalog service. It can be up to 600 characters long and include HTML formatting.
objectId	String	Guaranteed	The unique object ID of the catalog service.
objectUri	String	Guaranteed	The URI of the object.
objectUrl	String	Guaranteed	The URL of the object.
csCategoryName	String	Guaranteed	The category name of the catalog service.
csCategoryObjectId	String	Guaranteed	The unique object ID of the category.

Table 505. Fields for a catalog service object (continued)

Field	Type	Guaranteed/ Optional	Description
csState	String	Guaranteed	<p>The current state of the catalog service. Valid values are:</p> <p>published The service is published and available for users to run.</p> <p>pending-publish-approval The service is waiting for approvals for it to be published.</p> <p>incomplete The service is being created and tested, prior to a request to be published.</p> <p>suspended The service has been temporarily prevented from being run. This value is only applicable to published services.</p> <p>archived The service has been archived.</p> <p>approved All approvers listed in the service definition have approved the service.</p>
csCreatedTime	String	Guaranteed	The creation date and time of the catalog service. The value is in ISO 8601 format. For example, 2019-07-30T14:22:08.614Z
csCreatedBy	String	Guaranteed	The user ID that created the catalog service.
csDefaultInstanceExpiryInterval	String	Guaranteed	<p>This is defaulted to 0 if nothing was sent during the creation of the request. The time when the service expires, in ISO 8601 format. Services are automatically changed to the state "expired" when the current time exceeds the service creation time plus the service expiry.</p> <p>For example:</p> <ul style="list-style-type: none"> • P1D – One day • PT18H – Eighteen hours
csApprovals	List	Guaranteed	List of approval objects for the catalog service. If the approver is not yet set, an empty list is returned.
csExpiryUpdateableInRunner	boolean	Guaranteed	This is set to false if nothing was sent in the request. The value that is used to determine whether the user can override the expiry interval in the service runner.
csLastModifiedTime	String	Guaranteed	The date and time of the last modification to the catalog service, in ISO 8601 format.
csLastModifiedBy	String	Guaranteed	The user ID that last modified the catalog service.

Table 505. Fields for a catalog service object (continued)

Field	Type	Guaranteed/ Optional	Description
csBeingModifiedBy	String	Guaranteed	The ID of the user who is updating the service definition.
csOriginalWfDefinitionPath	String	Guaranteed	The absolute path of the original workflow definition file used to create the catalog service. The value cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
csOriginalWfVariablePropertiesPath	String	Optional	The absolute path of the optional original workflow variable properties file used to create the catalog service. The value cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
csPages	Array of csPage objects	Guaranteed	An object that acts as a container with the information that is related to the pages in the service runner. For more information, see Table 511 on page 971 .
csInputLogic	Array of csInputLogic	Guaranteed	Object that contains the logic mapping for the catalog service inputs. Each key in this object is the name of an input, which then contains a csControlInputDefinitionId string, and a csDependentLogics object. csControlInputDefinitionId Is a String that is the Inputs name. csDependentLogic objects are controlled by the csInputDefinition. For more information, see Table 510 on page 970 .
csInputDefinitions	Map of csInputDefinition objects with the input definition identifier as the key	Guaranteed	The Map of InputDefinitions with the key as input variable name. Each definition contains its own csInputValidation object. This represents all the required information provided for an input variable in the workflow definition file.
runAsUserSteps	Array of RunAsUserStep Objects	Guaranteed	The array of runAsUserStep objects. Each runAsUserStep object contains its own list of runAsUserStepApproval objects. For more information, see Table 513 on page 972 .
csRunOnAllSystems	boolean	Guaranteed	When true, the catalog service can run on any system found in the systems table list.
csTargetSystems	Array of Strings	Guaranteed	List of target systems on which the service is allowed to run.

Table 505. Fields for a catalog service object (continued)

Field	Type	Guaranteed/ Optional	Description
csChangeRecordRequiredSystems	Array of Strings	Guaranteed	List of target systems on which change record is always required.
csChangeRecordAlwaysRequired	boolean	Guaranteed	When true, the change record is required on all target system and csChangeRecordRequiredSystems is empty.
csActivityTimeline	Array of ActivityTimeline Objects	Guaranteed	List of objects that identifies the history of events pertaining to this catalog service. For more information, see Table 515 on page 973 .
csJobStatement	String	Guaranteed	The default JOB statement set of this catalog service. If none is set, an empty string is returned.
mgmtsvcsVersion	long	Guaranteed	Identifies the z/OS Management Services Catalog version of the service.
mgmtsvcsVersionSupported	boolean	Guaranteed	Indicates whether REST operations are allowed for the catalog service. Operations are not allowed if the version associated with the level of the z/OS Management Services plug-in is LESS than that of this object. <ul style="list-style-type: none"> • True if operations are allowed • False if operations are not allowed

Table 506. Fields for each catalog service input definition object (csInputDefinition)

Field	Type	Guaranteed/ Optional	Description
csInputWfVariable	String	Guaranteed	The name of the applicable workflow variable.
csInputLabel	String	Guaranteed	A short label that identifies the variable. It can be up to 50 characters. The value cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
csInputDescription	String	Optional	A detailed description of the variable. It can be up to 600 characters long and include HTML formatting.
csInputType	String	Guaranteed	"string" "boolean" "integer" "float" "date-time"

Table 506. Fields for each catalog service input definition object (csInputDefinition) (continued)

Field	Type	Guaranteed/ Optional	Description
csInputWidget	String	Guaranteed	<p>The UI widget used to obtain the value for the input variable. Valid values are:</p> <ul style="list-style-type: none"> • check-box • radio-button • drop-down • text-box • edit-box • combo-box • date-picker • time-picker • hard-coded-value <p>"check-box", "radio-button", "drop-down", "text-box", "edit-box", "combo-box", "date-picker", "time-picker", "hard-coded-value.</p> <p>The default widget is provided at create based on number of choices and the setting of csInputValueMustBeChoice.</p> <p>The detailed explanation is provided in a table below.</p>
csInputWfVariable Category	String	Guaranteed	Name of the workflow variables category designation from the workflow definition.
csInputValidation	Object of csInputValidation	Optional	Gives the ability to provide the input validation to apply to the input validation. For more information, see Table 512 on page 971 .
csInputValues	Array of csInputType	Optional	<p>The allowable selection of input values for the variable. The actual type of each of the values must be consistent with the csInputType for the input definition.</p> <p>These values override what is specified in the workflow definition file unless the workflow variable is restricted to only the choices specified in the workflow variable. When that occurs, this property is ignored.</p>
csInputValuesMetadata	Object of csInputValue Metadata	Optional	<p>Specifies metadata that is associated with each of the entries for the csInputValues property. For more information, see Table 507 on page 969.</p> <p>The property name of each entry is the corresponding value from the csInputValues property.</p>

Table 506. Fields for each catalog service input definition object (csInputDefinition) (continued)

Field	Type	Guaranteed/ Optional	Description
csInputDefault	String	Optional	The default value provided in the workflow definition file.
csInputRequiredAt Create	boolean	Guaranteed	Identifies whether the input value is required to be specified when you submit the service.
csInputPromptAt Create	boolean	Guaranteed	Identifies whether the input value should be explicitly prompted for when submitting the service.
csInputIncludePrefix	boolean	Guaranteed	This property reflects information from the workflow definition and is not currently used by z/OS Management Services Catalog.
csInputValueMust BeChoice	boolean	Guaranteed	For a string type variable, this property specifies whether the value must be one of the given choice values. The value is determined by the workflow definition and cannot be changed. When csInputValueMustBeChoice is true, csRestrictInputValueChoices is true and cannot be changed.
csRestrictInputValueChoices	boolean	Optional	This property controls the input widget behavior for string type variables that have multiple input value choices that are specified by the workflow definition. When the workflow definition specifies the input values must be restricted to the specified choices by setting csInputValueMustBeChoice to true, the value for csRestrictInputValueChoices is set to true and cannot be changed. If csInputValueMustBeChoice is false, csRestrictInputValueChoices can be set to true to prevent the use of ad-hoc user-supplied values that the workflow definition would otherwise allow.
csInputFilePathVariable	boolean	Guaranteed	This property identifies whether the csInputDefinition is used for file path substitution. If true, the input definition must be assigned a hard-coded-value as part of a service creation. Once the service is created, this value cannot be changed.

Table 506. Fields for each catalog service input definition object (csInputDefinition) (continued)

Field	Type	Guaranteed/ Optional	Description
csInputErrorMessage	String	Guaranteed	The error message to be displayed if the value entered for the variable is not valid. If there is no error message specified in the workflow definition file, the default message is The value entered is not valid.
csInputRequired	boolean	Guaranteed	Specifies whether the input definition is required when you submit the service. For dependent inputs, this value is honored only when the logic conditions of the control variable are met.

Table 507. Fields for the csInputValueMetadata object

Field	Type	Guaranteed/ Optional	Description
csChoiceValueDefinedByWorkflow	boolean	Guaranteed	Specifies whether the corresponding csInputValues entry is defined by the workflow definition.

Table 508 on page 969 defines Default widget selection for the String Inputs.

Table 508. z/OS Management Services Catalog default widget selection for the string inputs

Choices	Number of Choices	Required/ Optional	Values must be choice	Multi-line	Default widget
No	N/A	Required/ Optional	N/A	False	TEXT_BOX
No	N/A	Required/ Optional	N/A	True	TEXT_AREA
Yes	1	Required	True	N/A	HARD_CODED_VALUE
Yes	1	Required	False	N/A	RADIO_BUTTON
Yes	1	Optional	True	N/A	CHECK_BOX
Yes	1	Optional	False	N/A	RADIO_BUTTON
Yes	2-7	Required/ Optional	True/False	N/A	RADIO_BUTTON
Yes	8+	Required/ Optional	True/False	N/A	DROP_DOWN

Table 509. Fields for each catalog service input logic object (csInputLogic)

Field	Type	Guaranteed/Optional	Description
csControlInputDefinitionId	String	Guaranteed	The csInputWfVarName name of the csInputDefinition that controls whether a dependent csInputDefinition object is displayed in the service runner.
csDependentLogics	Array of csDependentLogic	Guaranteed	The array of csDependentLogic objects controlled by the csInputDefinition For more information, see Table 510 on page 970 .

Table 510. Fields for each dependent logic object (csDependentLogic)

Field	Type	Guaranteed/Optional	Description
csControlInputDefinitionComparator	String	Guaranteed	The logical comparator to apply to the selected value of the control variable when attempting to run the service. Valid logic operations depend on the csControlVariableType. Valid logic operations are: <ul style="list-style-type: none"> • equal • not-equal • true • false • greater • less • greater-or-equal • less-or-equal • not-null
csControlInputDefinitionComparatorValue	String	Guaranteed	The value of the controlling input definition that triggers this csInputLogicobject. The value is in string format and will be normalized based on the csControlVariableType.
csControlInputDefinitionComparatorValue	String	Guaranteed	The csInputWfVarName of the input definition whose behavior is determined by this csInputLogic Object.

Table 511. Fields for each catalog service page object (csPage)

Field	Type	Guaranteed/ Optional	Description
csPageName	String	Guaranteed	Name of the input page. It can be up to 50 characters. The name cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
csPageDescription	String	Optional	The description for the input page. It can be up to 600 characters long and include HTML formatting.
csInputOrder	Array of Strings	Guaranteed	An array used to specify the order of the input definitions that is displayed in the service builder/service runner. These input definitions must match the csInputWfVarName values from the csInputDefinitions list.

Table 512. Fields for Catalog Service Input Definition Validation (csInputValidation)

Field	Type	Guaranteed/ Optional	Description
minLength	Integer	Optional	Minimum length required
maxLength	Integer	Optional	Maximum length allowed
minLongValue	long	Optional	Minimum long value required
maxLongValue	long	Optional	Maximum long value allowed
minDecimalValue	double	Optional	Minimum decimal value allowed
maxDecimalValue	double	Optional	Maximum decimal value allowed
decimalPlaces	Integer	Optional	Number of decimal places allowed for a float value
maxDate	Date	Optional	Maximum date allowed (yyyy-mm-dd)
minDate	Date	Optional	Minimum date allowed (yyyy-mm-dd)
maxTime	Time	Optional	Maximum time allowed (hh:mm:ss)
minTime	Time	Optional	Minimum time allowed (hh:mm:ss)
regularExpression	String	Optional	Regular Expression that can be used to validate the input.

Table 512. Fields for Catalog Service Input Definition Validation (csInputValidation) (continued)

Field	Type	Guaranteed/ Optional	Description
validationType	String	Optional	<p>Enum of ValidationType. The expected values are:</p> <ul style="list-style-type: none"> • ALPHA • ALPHAB • ALPHANUM • BIT • DSMEMBERNAME • DSNAME • DSQUAL • GROUP • HEX • IPADDR • IPADDR4 • IPADDR6 • TSouserID • UNIXID • USERID • VOLSER <p>Every validationType comes with the equivalent regular expression.</p>

Table 513. Fields for Catalog Service RunAsUser Step (runAsUserSteps)

Field	Type	Guaranteed/ Optional	Description
stepName	String	Guaranteed	Name of the step
stepTitle	String	Guaranteed	The title of the step that needs to be displayed.
stepDescription	String	Optional	The description of the step.
stepRunAsUser	String	Guaranteed	The ID of the user who is listed as the RunAsUser of this step.
stepRunAsUserSubstitutable	boolean	Guaranteed	Boolean value that defines whether the run as user has any substitutions.
stepApprovals	Array of stepRunAsUserApprovals	Guaranteed	Contains information about all the approvers that are listed for this step. For more information, see Table 514 on page 973 .

Table 514. Fields for Catalog Services RunAsUser Step Approvals (stepApprovals)

Field	Type	Guaranteed/ Optional	Description
stepRunAsUserApprovers	Array of Strings	Guaranteed	One or more approvers are allowed. This is a list of all the approvers that are listed for this step in the workflow definition file.
stepRunAsUserApproverSubstitutable	Boolean	Guaranteed	Boolean value that defines whether the run as user has any substitutions.

Table 515. Fields for an ActivityTimeLine object

Field	Type	Guaranteed/ Optional	Description
actionName	String	Guaranteed	Name of the action that is performed.
actionComment	String	Optional	Comment provided by the user performing the action.
actionTime	String	Guaranteed	When the action was performed.
actionUser	String	Guaranteed	The user who performed the action.
actionAdditionalInfo	Object	Guaranteed	Object that contains any potential additional information about the action.

Example HTTP interaction

The example in Figure 440 on page 973 shows a request to the details of a service in the z/OS Management Services catalog.

```
GET /zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705
```

Figure 440. Sample request to get the details of a specific service in the z/OS Management Services catalog

The following example shows an example response body that contains the details of the catalog service.

```
{
  "csName": "Great New Service",
  "csDescription": "A description of this great new services.",
  "objectId": "128c778f-94444030-b2d13509-87640705",
  "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
  "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
  "csCategoryObjectId": "128c778f-94444030-b2d13509-87640705",
  "csCategoryName": "Catalog Management",
  "csState": "published",
  "csCreatedTime": "2019-11-15T19:12:45.223Z",
  "csDefaultInstanceExpiryInterval": "PT98H",
  "csCreatedBy": "suser1",
  "csLastModifiedTime": "2019-12-22T09:12:48.291Z",
  "csLastModifiedBy": "suser2",
  "csOriginalWfDefinitionPath": "/u/suser1/workflows/createZosThing.xml",
  "csOriginalWfVariablePropertiesPath": "/u/suser1/workflows/wfVariable.properties",
  "mgmtsvcsVersion": 100,
  "mgmtsvcsVersionSupported": true,
  "csExpiryUpdateableInRunner": "false",
  "csTargetSystems": ["sys1", "sys2", "sys3"],
  "csRunOnAllSystems": true,
  "csChangeRecordAlwaysRequired": false,
  "csChangeRecordRequiredSystems": ["sys1", "sys2", "sys3"],
}
```

```

        "csApprovals": [
            {
                "csApprovalId": "e821ee1-2b4b483-926728e-fb6114f",
                "csApprovers": [
                    "ibmuser"
                ],
                "csApprovalIsRunAsUser": false,
                "csApprovalStatus": "PENDING"
            }
        ],
        "csBeingModifiedBy": "user21",
        "csPages": [{
            "csPageName": "Untitled",
            "csPageDescription": "Detailed information about this page",
            "csInputOrder": [
                "remountOption",
                "fileSystemName"
            ]
        }],
        "csInputDefinitions": {
            "remountOption": {
                "csInputWfVarName": "remountOption",
                "csInputLabel": "Remount Option",
                "csInputDescription": "This variable determines the behavior of the remounting
process",
                "csInputType": "string",
                "csInputWfVariableCategory": "Configuration",
                "csInputErrorMessage": "The value entered is not valid",
                "csInputFilePathVariable": false,
                "csInputValidation": {
                    "maxLength": 100000
                },
                "csInputValues": [
                    "SWITCH",
                    "RDRW",
                    "READ",
                    "SAMEMODE"
                ],
                "csInputValuesMetadata": {
                    "SWITCH": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "RDRW": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "READ": {
                        "csChoiceValueDefinedByWorkflow": true
                    },
                    "SAMEMODE": {
                        "csChoiceValueDefinedByWorkflow": true
                    }
                },
                "csInputValueMustBeChoice": true,
                "csInputWidget": "radio-button",
                "csInputIncludePrefix": true,
                "csInputDefault": "SWITCH",
                "csInputRequiredAtCreate": false,
                "csInputPromptAtCreate": true,
                "csInputRequired": false
            },
            "fileSystemName": {
                "csInputWfVarName": "fileSystemName",
                "csInputLabel": "File System Name",
                "csInputDescription": "The fully qualified name of the date set containing the file
system",
                "csInputType": "string",
                "csInputWfVariableCategory": "Configuration",
                "csInputErrorMessage": "The value entered is not valid",
                "csInputFilePathVariable": false,
                "csInputValidation": {
                    "regularExpression": "[A-Z#$$@]",
                    "validationType": "DSNAME"
                },
                "csInputValues": [],
                "csInputValuesMetadata": {},
                "csInputWidget": "text-box",
                "csInputIncludePrefix": true,
                "csInputRequiredAtCreate": false,
                "csInputPromptAtCreate": true,
                "csInputRequired": false
            }
        }
    ],
    {

```

```

    "csInputLogic": {
      "unmountOption": {
        "csControlInputDefinitionId": "unmountOption",
        "csDependentLogics": [
          {
            "csControlInputDefinitionComparatorValues": [
              "REMOUNT"
            ],
            "csControlInputDefinitionComparator": "equal",
            "csDependentInputDefinitionIds": [
              "remountOption"
            ]
          }
        ]
      }
    },
    "csRunAsUserSteps": [
      {
        "stepName": "Step1",
        "stepTitle": "Step 1",
        "stepDescription": "Brief description about the step",
        "stepRunAsUser": "usera",
        "stepRunAsUserSubstitutable": false,
        "stepApprovals": [
          {
            "stepRunAsUserApprovers": ["apprId1", "apprId2"],
            "stepRunAsUserApproverSubstitutable": false
          }
        ]
      },
      {
        "stepName": "step2",
        "stepTitle": "Step 2",
        "stepDescription": "Brief description about the step",
        "stepRunAsUser": "userb",
        "stepRunAsUserSubstitutable": false,
        "stepApprovals": [
          {
            "stepRunAsUserApprovers": ["apprId2"],
            "stepRunAsUserApproverSubstitutable": false
          }
        ]
      }
    ],
    "csActivityTimeline": [{
      "actionName": "REQUEST-PUBLISH-APPROVAL",
      "actionComment": null,
      "actionTime": "2020-07-09T20:17:56.801Z",
      "actionUser": "zosmfad",
      "actionAdditionalInformation": null
    }]
  }
}

```

Get the list of categories

This operation gets the list of categories that are defined in z/OS Management Services (zMSC).

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/categories
```

Query parameters

None.

Description

This operation returns the list of categories that are defined to z/OS Management Services.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request returned the list of categories. A response body is provided, as described in [“Response content” on page 976](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or zMSC Administrator.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in ["Response content" on page 976](#).

For more information about error status codes, see ["HTTP status codes" on page 955](#).

Response content

On successful completion, the service returns a response body, which contains a list of Category JSON object.

For more information about the definition of the Category JSON object, see [Table 516 on page 976](#).

Table 516. Fields for a Category object			
Field	Type	Guaranteed/ Optional	Description
objectId	String	Guaranteed	The unique object ID of the category
objectUri	String	Guaranteed	The URI of the object.
objectUrl	String	Guaranteed	The URL of the object.
ccName	String	Guaranteed	The category's name. (Limit of 50 characters).
ccDescription	String	Optional	An optional description for the category. (Limit of 250 characters).
ccCreatedTime	String	Guaranteed	The creation date and time of the catalog service. The value is in ISO 8601 format. For example: 2019-07-30T14:22:08.614Z.
ccCreatedBy	String	Guaranteed	The user ID that created the catalog service.
ccLastModifiedTime	String	Guaranteed	The date and time of the last modification to the catalog service, in ISO 8601 format.
ccLastModifiedBy	String	Guaranteed	The user ID that last modified the catalog service.
mgmtsvcsVersion	String	Guaranteed	Identifies the management services version of the catalog service.

Table 516. Fields for a Category object (continued)

Field	Type	Guaranteed/ Optional	Description
mgmtsvcsVersionSupported	boolean		<p>Indicates whether GET, POST, PUT, and DELETE operations are allowed for the catalog service. Operations are not allowed if the version associated with the level of the z/OS Management Services plug-in is LESS than that of this object.</p> <ul style="list-style-type: none"> • true if operations are allowed • false if operations are not allowed

Example HTTP interaction

The example in Figure 441 on page 977 shows a request to get the list of categories defined in z/OS Management Services.

```
GET /zosmf/mgmt-services/rest/categories
```

Figure 441. Sample request to get the list of categories defined to z/OS Management Services.

The following example shows a response body that contains a list of categories.

```
[
  { "objectId": "128c78f-9444030-b2d3509-8764705",
    "ccName": "zFS Services",
    "ccDescription": "Services to manage zFS datasets.",
    "objectUri": "/zosmf/mgmt-services/rest/categories/128c78f-9444030-b2d13509-87640705",
    "objectUrl": "https://testbed.ibm.com/zosmf/mgmt-services/rest/categories/128c78f-9444030-b2d13509-87640705",
    "ccCreatedTime": "2019-12-05T02:41:55.720Z",
    "ccCreatedBy": "ibmuser",
    "ccLastModifiedTime": "2019-12-05T02:41:55.720Z",
    "ccLastModifiedBy": "ibmuser",
    "mgmtsvcsVersion": 100,
    "mgmtsvcsVersionSupported": true},
  { "objectId": "9d4cce8-91f8435-978e3a6-5632c6a",
    "ccName": "RACF Userid Services",
    "ccDescription": "Services to deal with RACF user IDs.",
    "objectUri": "/zosmf/mgmt-services/rest/categories/9d4cce8-91f8435-978e3a6-5632c6a",
    "objectUrl": "https://testbed.ibm.com/zosmf/mgmt-services/rest/categories/9d4cce8-91f8435-978e3a6-5632c6a",
    "ccCreatedTime": "2019-12-05T01:41:29.665Z",
    "ccCreatedBy": "ibmuser",
    "ccLastModifiedTime": "2019-12-05T01:41:29.665Z",
    "ccLastModifiedBy": "ibmuser",
    "mgmtsvcsVersion": 100,
    "mgmtsvcsVersionSupported": true},
  { "objectId": "de6d087-3013446-9068922-aa2770c",
    "ccName": "z/OS Catalog Services",
    "ccDescription": "Services to manage z/OS catalogs.",
    "objectUri": "/zosmf/mgmt-services/rest/categories/de6d087-3013446-9068922-aa2770c",
    "objectUrl": "https://localhost:4444/zosmf/mgmt-services/rest/categories/de6d087-3013446-9068922-aa2770c",
    "ccCreatedTime": "2019-12-05T01:42:38.325Z",
    "ccCreatedBy": "ibmuser",
    "ccLastModifiedTime": "2019-12-05T01:42:38.325Z",
    "ccLastModifiedBy": "ibmuser",
    "mgmtsvcsVersion": 100,
    "mgmtsvcsVersionSupported": true}
]
```

Get category details

This operation gets the details for a category that is identified by its object identifier.

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/categories/{object-id}
```

In this request:

{object-id}

Identifies the category that is retrieved.

Query parameters

None.

Description

This operation returns the details of a specific category.

On successful completion, HTTP status code 200(OK) is returned, indicating that the request returned the details of the requested category. A response body is provided as, described in [“Response content” on page 976](#).

Request content

None.

Authorization requirements

The user’s z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or zMSC Administrator.

Status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 976](#).

For more information about error status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service returns a response body, which contains a list of Category JSON object.

For more information about the definition of the Category JSON object, see [Table 516 on page 976](#).

Example HTTP interaction

The example in [Figure 442 on page 978](#) shows a request to the details of a z/OS Management Services category.

```
GET /zosmf/mgmt-services/rest/categories/5adae71-66be402-94a1b89-f80613e
```

Figure 442. Sample request to get the details of a specific category defined to z/OS Management Services

The following example shows the response body that contains the details of the requested category.

```
{
  "objectId" : "5adae71-66be402-94a1b89-f80613e",
  "ccName" : "IOCDs Services",
  "ccDescription" : "Services to do various things with IOCDs.",
  "objectUri" : "/zosmf/mgmt-services/rest/categories/5adae71-66be402-94a1b89-f80613e",
  "objectUrl" : "https://testbed.ibm.com/zosmf/mgmt-services/rest/categories/5adae71-66be402-94a1b89-f80613e",
  "ccCreateTime" : "2019-12-05T02:41:55.720Z",
  "ccCreatedBy" : "ibmuser",
  "ccLastModifiedTime" : "2019-12-05T02:41:55.720Z",
  "ccLastModifiedBy" : "ibmuser",
  "mgmtsvcsVersion" : 100,
  "mgmtsvcsVersionSupported" : true
}
```

List the service submissions

Use this operation to list the service submissions that are created in z/OS Management Services (zMSC).

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/service-instances
```

Query parameters

The query parameters are exclusive except for the 'summary' parameter. For instance, `/zosmf/mgmt-services/rest/service-instances?label=ProjectA&summary=true` returns a summary for all the service submission that have the label ProjectA.

Table 517. Query parameters		
Query Parameter	Required/Optional	Description
serviceName	Optional	A Regular Expression. Specifies the name of the service submission.
status	Optional	All service submissions in the given state. States include: <ul style="list-style-type: none">• queued• running• completed• failed• terminated• draft
submitter	Optional	User ID of the user who submitted the service submission.
targetSystem	Optional	System that the service submission is targeted to run on.
label	Optional	Returns all service submissions with the given label.

Table 517. Query parameters (continued)

Query Parameter	Required/Optional	Description
Summary	Optional	<p>Boolean value. If nothing is provided, then it is assumed as false by default. Specifies to return a limited set of properties each catalog service submission.</p> <p>Summary Properties are:</p> <ul style="list-style-type: none"> • siName • objectId • objectUri • objectUrl • siStatus • siTargetSystem • siCreatedBy • siRunAfter • siExpires • siCatalogServiceDefinition • mgmtsvcsVersion • mgmtsvcsVersionSupported • siCatalogServiceDeleted

Description

This operation retrieves the list of service submissions that are created in z/OS Management Services.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request returned the list of service submissions. A response body is provided, as described in [Table 518 on page 981](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or zMSC Administrator.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned, and the response body is provided as described in [Table 518 on page 981](#). If no results are returned from the specified query parameters, HTTP status code 204 (No Content) is returned.

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service submission returns a response body that contains a JSON object with details about the service submissions. For more information, see [Table 518 on page 981](#). The service submission details only contain a summary of their service definitions when they are returned from this request. This is different from the full service definition that is returned when only a single service submission is returned from a GET request.

Table 518. Array of objects	
Type	Description
Array of service submission objects. For more information, see Table 522 on page 991 .	Array of service submission objects. The array is filtered based on any query parameters that were provided.

Example HTTP interaction

The example in [Figure 443 on page 981](#) shows a request to get the list of service submissions in the z/OS Management Services.

```
GET /zosmf/mgmt-services/rest/service-instances
```

Figure 443. Sample request to get the list of service submissions in z/OS Management Services Catalog

The following example shows a response body that contains a single service submission.

```
[
  {
    "siName": "Great New Service",
    "objectId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
    "objectUri": "/zosmf/mgmt-services/rest/service-instances/ aaaaaaaaa-aaaaaaaa-aaaaaaaa-
aaaaaaab",
    "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/service-instances/
aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
    "siCreatedBy": "ConnerF",
    "siCreatedTime": "2019-12-10T09:10:43.221Z",
    "siLastModifiedTime": "2019-12-20T09:10:43.221Z",
    "siLastModifiedBy": "ConnerF",
    "siCatalogServiceId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaabb",
    "siCatalogServiceDeleted": false,
    "siQueuedTime": "2019-12-20T09:10:40.221Z",
    "siStatus": "Queued",
    "siTargetSystem": "PLEX1.TEST",
    "siRunAutomatically": false,
    "siRunAfter": "2019-12-22T09:12:48.291Z",
    "siExpires": "2019-12-24T09:12:48.291Z",
    "siChangeId": "A180418",
    "siBeingModifiedBy": "VC",
    "siActivityTimeline": [
      {
        "siActionName": "Modified",
        "siActionUser": "ConnerF",
        "siActionComment": "I changed something",
        "siActionTime": "2019-12-20T09:10:20.221Z"
      },
      {
        "siActionName": "Added to queue",
        "siActionUser": "ConnerF",
        "siActionTime": "2019-12-20T09:10:40.221Z"
      }
    ],
    "siInputs": {
      "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "SMS Managed",
      "baaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "SMS252",
      "daaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "Some variable value",
      "eaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": null
    },
    "siPreviousInputs": {
      "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "NonSMSManaged"
    },
    "mgmtsvcsVersion": 100,
    "mgmtsvcsVersionSupported": true
  },
  {
  }
```

```

        "siName": "Other Great New Service",
        "objectId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaac",
        "objectUri": "/zosmf/mgmt-services/rest/service-instances/ aaaaaaaa-aaaaaaaa-aaaaaaaa-
aaaaaac",
        "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/service-instances/
aaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaac",
        "siCreatedBy": "ConnerF",
        "siCreatedTime": "2019-12-10T09:10:43.221Z",
        "siLastModifiedTime": "2019-12-21T09:10:43.221Z",
        "siLastModifiedBy": "ConnerF",
        "siRunTime": "2019-12-20T09:10:40.221Z",
        "siCatalogServiceId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaabbb",
        "siCatalogServiceDeleted": false,
        "siWorkflowInstanceName": "MGMSVS-Great New Service-1",
        "siWorkflowInstanceKey": "63c12089-1267-413d-a671-bb187e1e76db",
        "siStatus": "Running",
        "siRunAutomatically": true,
        "siTargetSystem": "PLEX1.TEST",
        "siRunAfter": "2019-12-23T09:12:48.291Z",
        "siExpires": "2019-12-26T09:12:48.291Z",
        "siChangeId": "A180418",
        "siBeingModifiedBy": "VC",
        "siJobStatement": "",
        "siActivityTimeline": [
            {
                "actionName": "COMMENT",
                "actionComment": "hello world",
                "actionTime": "2020-07-09T20:11:59.775Z",
                "actionUser": "zosmfad",
                "actionAdditionalInformation": null
            }
        ],
        "siInputs": {"aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaa5": "Some Variable Value"},
        "siCatalogServiceDefinition": {
            "csName": "Great New Service",
            "objectId": "128c778f-94444030-b2d13509-87640705",
            "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-
b2d13509-87640705",
            "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-
services/128c778f-94444030-b2d13509-87640705",
            "csCategory": "Catalog Management",
            "csState": "published",
            "csTags": [ "tag1", "tag2" ],
            "mgmtsvcsVersion": 100,
            "mgmtsvcsVersionSupported": true,
            "mgmtsvcsVersion": 100,
            "mgmtsvcsVersionSupported": true
        }
    ]
}

```

Create a new service submission

Use this operation to create a new draft service submission in z/OS Management Services Catalog (zMSC) task.

HTTP method and URI path

POST /zosmf/mgmt-services/rest/service-instances

Query parameters

None.

Description

This operation creates a new service submission in z/OS Management Services Catalog.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new service submission.

Request content

A request body is required for this request. For more information, see [Table 519 on page 983](#).

Table 519. Fields for a service submission create object			
Field	Type	Required/Optional	Description
siCatalogServiceId	String	Required	UUID of the catalog service that the created service submission is being created from.
siTargetSystem	String	Optional	System that the service submission is run on.
siRunAutomatically	Boolean	Optional	If true, the service submission can run automatically. If excluded or false, the service submission must be run manually.
siChangeRecord	String	Optional	String that allows the user to associate the new service submission with a change record identifier from an external change management system.
siComment	String	Optional	String that allows the service submission creator to enter a custom comment to associate with the newly created service submission.
siRunAfter	String	Optional	The service submission cannot be run before the specified time.
siExpires	String	Optional	The service submission cannot be run before the specified time.
siInputs	JSONObject	Optional	An object that identifies workflow variable names as the property keys and their respective values. These are used to create the workflow instance when the service submission runs. Each key's corresponding value varies.

Table 519. Fields for a service submission create object (continued)

Field	Type	Required/Optional	Description
siJobstatement	String	Optional	The JOB statement to use when the submission runs. If siJobstatement is not specified, a default JOB statement is used.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or a zMSC Administrator.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security configuration requirements for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned and a response body is provided, as described in [“Response content” on page 984](#).

For more information about HTTP status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service submission returns a response body, which contains a JSON object with details about the domain. See [Table 520 on page 984](#).

Table 520. Fields for a get service submission object

Field	Type	Guaranteed/Optional	Description
siName	String	Guaranteed	Name of the service submission. This is the same as the name of the catalog service that the service submission was created from.
objectId	String	Optional	The unique object ID of the service submission.
objectUri	String	Guaranteed	The URI of the service submission.
objectUrl	String	Guaranteed	The URL of the service submission.
siCreatedBy	String	Guaranteed	User who creates the service submission.
siCreatedTime	String	Guaranteed	The time when the service submission was first created.
siLastModifiedTime	String	Guaranteed	The time the service submission was last modified. If no modifications occur, this is the time that the service submission was created.
siLastModifiedBy	String	Guaranteed	The last user to modify this service submission.
siQueuedTime	String	Optional	The time when the service submission enters the queued status.
siRunTime	String	Optional	The time when the service submission enters the queued status.

Table 520. Fields for a get service submission object (continued)

Field	Type	Guaranteed/ Optional	Description
siStatus	String	Optional	Status of the service submission Status include: <ul style="list-style-type: none"> • Queued • Running • Failed • Completed • Terminated • Draft
siTargetSystem	String	Guaranteed	System that the service submission is running on.
siRunAutomatically	Boolean	Guaranteed	If true, the service submission can run automatically. If false, the service submission must be run manually.
siWorkflowInstanceName	String	Optional	If the service submission has started, the workflow name in the workflows task that is created to run the service submission.
siWorkflowInstanceKey	String	Optional	If the service submission has started, the workflow key in the workflows task that was created to run the service submission.
siCatalogServiceId	String	Guaranteed	The UUID of the category service submission that the created service submission was created from.
siCatalogServiceDeleted	Boolean	Guaranteed	Indicates whether or not the underlying catalog service submission is deleted.
siRunAfter	String	Optional	The service submission cannot be run before this time if it is returned.
siExpires	String	Optional	The service submission cannot be run before this time if it is returned.
siChangeRecord	String	Optional	The value that the user enters. Intended to be associated with an external change management system.
siActivityTimeline	JSONArray of ActivityTimeline Objects	Guaranteed	List of objects that display the history of events that pertains to this service submission. For more information, see Table 521 on page 986 .
siInputs	JSONObject	Optional	JSONObject with a key value pair of the variable ID and its value.
siPreviousInputs	JSONObject	Optional	JSONObject with key value pair of variable values and the value it was set to before its current value.
mgmtsvcsVersion	String	Guaranteed	Identifies the management services version of the catalog service.

Table 520. Fields for a get service submission object (continued)

Field	Type	Guaranteed/Optional	Description
mgmtsvcsVersionSupported	Boolean	Guaranteed	Indicates whether GET, POST, PUT, and DELETE operations are allowed for the catalog service. Operations are not allowed if the version associated with the level of the z/OS Management Services plug-in is LESS than this object. <ul style="list-style-type: none"> • "True" if operations are allowed. • "False" if operations are not allowed.
siCatalogServiceDefinition	JsonObject	Optional	JSON Object that contains the service submission's catalog service definition. For more information, see Table 505 on page 963 . If the catalog service gets deleted, this property is not returned and siCatalogServiceDeleted is set to true.
siBeingModifiedBy	String	Guaranteed	User who is modifying the service submission. Other users must use the Acquire lock action to be able to update the service submission.
siJobStatement	String	Guaranteed	The JOB statement that is sent to the workflow engine when the service submission is run. If the string is empty, the workflow engines default JOB statement is used.
siTestRun	Boolean	Guaranteed	"True" if the service submission is a test run. "False" if it is a standard run.

Table 521. Fields for an ActivityTimeLine object

Field	Type	Guaranteed/Optional	Description
actionName	String	Guaranteed	Name of the action that is performed.
actionComment	String	Optional	Comment provided by the user that performs the action.
actionTime	String	Guaranteed	When the action is performed.
actionUser	String	Guaranteed	The user who performed the action.
actionAdditionalInfo	Object	Guaranteed	Object that contains any potential additional information about the action.

Example HTTP interaction

The example in [Figure 444 on page 987](#) shows a request to create a new service submission in the z/OS Management Services. This response body contains the UUID of the newly created service submission.


```
POST /zosmf/mgmt-services/rest/service-instances
{
  "siCatalogServiceId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa",
  "siTargetSystem": "PLEX5.TEST",
  "siRunAuto": false,

  "siChangeRecord": "A180999",
  "siComment": "Starting a new low priority service for project A to do a thing",
  "siRunAfter": "2020-1-24T09:12:48.291Z",
  "siExpire": "2020-1-29T09:12:48.291Z"
}
```

Figure 444. Sample request to create a new service submission in the z/OS Management Services Catalog

The example in [Figure 445 on page 987](#) shows a request to the details of a service submission in z/OS Management Services Catalog.

```
GET /zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab
```

Figure 445. Sample request to get the details of a specific service submission in the z/OS Management Services

The following example shows a response body that contains the details of a service submission.

```
{
  "siName": "Great New Service",
  "objectId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "objectUri": "/zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "siCreatedBy": "ConnerF",
  "siCreatedTime": "2019-12-10T09:10:43.221Z",
  "siLastModifiedTime": "2019-12-20T09:10:43.221Z",
  "siLastModifiedBy": "ConnerF",
  "siCatalogServiceId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "siCatalogServiceDeleted": false,
  "siQueuedTime": "2019-12-20T09:10:40.221Z",
  "siStatus": "Queued",
  "siTargetSystem": "PLEX1.TEST",
  "siRunAutomatically": false,
  "siRunAfter": "2019-12-22T09:12:48.291Z",
  "siExpires": "2019-12-24T09:12:48.291Z",
  "siChangeRecord": "A180418",
  "siBeingModifiedBy": "VC",
  "siJobStatement": "",
  "siTestRun": false,
  "siActivityTimeline": [
    {
      "actionName": "SUBMIT",
      "actionComment": null,
      "actionTime": "2020-07-09T20:18:54.441Z",
      "actionUser": "zosmfad",
      "actionAdditionalInformation": null
    }
  ],
  "siInputs": {
    "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa": "SMS Managed",
    "baaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa": "SMS252",
    "daaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa": "Some variable value",
    "eaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa": null
  },
  "siPreviousInputs": {
    "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaa": "NonSMSManaged"
  },
  "siCatalogServiceDefinition": {
    "csName": "Great New Service",
    "csDescription": "A description of this great new services.",
    "objectId": "128c778f-94444030-b2d13509-87640705",
    "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "csCategory": "Catalog Management",
    "csState": "published",
    "createdTime": "2019-11-15T19:12:45.223Z",
    "csTags": [ "tag1", "tag2" ]
  }
}
```

```

        "csDefaultInstanceExpiryInterval" : "PT98H",
        "csCreatedBy" : "suser1",
        "csLastModifiedTime" : "2019-12-22T09:12:48.291Z",
        "csLastModifiedBy" : "suser2",
        "csOriginalWfDefinitionPath" : "/u/suser1/workflows/createZosThing.xml",
        "csOriginalWfVariablePropertiesPath" : "/u/suser1/workflows/
wfVariable.properties",
        "mgmtsvcsVersion" : 0100,
        "mgmtsvcsVersionSupported" : 0100,
        "csExpiryUpdateableInRunner" : "false",
        "csPages" : [{
            "csPageName" : "page 1",
            "csPageDescription" : "Detailed information about this page",
            "csInputDefinitions" : {
                "createZosThingWfVar1": {
                    "csInputLabel" : "The first required
input",
                    "csInputDescription" : "Detailed information about this
input.",
                    "csInputType" : "string",
                    "csInputWidget" : "checkbox",
                    "csInputValues" : ["value1", "value2", "value3", "value4"],
                    "csInputWfVarName" : "createZosThingWfVar1",
                    "csInputValidation" : {
                        "minLength" : "20",
                        "maxLength" : "60"
                    }
                },
                "createZosThingWfVar2": {
                    "csInputLabel" : "The second input",
                    "csInputDescription" : "This input is required if Input
1 has the value DATASET.",
                    "csInputType" : "integer",
                    "csInputWidget" : "radio-button",
                    "csInputWfVarName" : "createZosThingWfVar2",
                    "csInputValidation" : {
                        "minLength" : "20",
                        "maxLength" : "60"
                    }
                },
                "createZosThingWfVar2a": {
                    "csInputLabel" : "Depends on Input 2",
                    "csInputDescription" : "This input only available when
Input 2 is false, and then it is optional.",
                    "csInputType" : "boolean",
                    "csInputWidget" : "radio-button",
                    "csDefaultValue" : true,
                    "csInputWfVarName" : "createZosThingWfVar2a"
                }
            }
        },
        "csInputOrder": ["createZosThingWfVar1", "createZosThingWfVar2a"]
    }],
    "csInputLogic": [
        {
            "csControlInputDefinitionId": "createZosThingWfVar2a",
            "csControlVariableLogiccsControlVariableComparator": "equal",
            "csControlVariableComparatorValue": "DATASET",
            "csDependentVariableId": "Input 2",
            "csDependentVariableRequired": true
        },
        {
            "csControlInputDefinitionId": "createZosThingWfVar1",
            "csControlVariableComparator": "equal",
            "csControlVariableComparatorValue": false,
            "csDependentVariableId": "Input 2a",
            "csDependentVariableRequired": false
        }
    ]
}

"mgmtsvcsVersion": 100,
"mgmtsvcsVersionSupported": true
}

```

Create a submission to run at the time you submit it

1. Create a new service submission draft with the POST `/zosmf/mgmt-services/service-instances` API.

In the request body:

- Issue the create request with **siRunAutomatically** set to **true**.
 - Specify values for the inputs in the **siInputs** property.
- Note:** Do not include the **siExpires** and **siRunsAfter** properties if you want to run the submission when you submit it.
- The **siTargetSystem** is optional and defaults to the z/OSMF host system unless otherwise specified.
 - The **siJobStatement** property is optional and can be used to specify a specific JOB statement when the submission runs. If **siJobStatement** is not specified, a default JOB statement is used.
2. Issue the **submit** action to run the draft submission using the POST `/zosmf/mgmt-services/service-instances/{submission-id}/actions/submit` API. The submission-id is included in the response from the Create new submission API.

Create a submission to manually run during a specified run window

1. Create a new service submission draft with the POST `/zosmf/mgmt-services/service-instances` API.

In the request body:

- Issue the create request with **siRunAutomatically** set to **false**.
 - Specify values for the **siRunAfter** and **siExpires** properties to establish the run window for the submission.
 - You must include the **siInputs** property and specify all of the necessary input values.
 - The **siTargetSystem** is optional and defaults to the z/OSMF host system unless otherwise specified.
 - The **siJobStatement** property is optional and can be used to specify a specific JOB statement when the submission runs.
2. Issue the **submit** action to submit and queue the draft submission using the POST `/zosmf/mgmt-services/service-instances/{submission-id}/actions/submit` API. The submission-id is included in the response from the Create new submission API. The queued submission is ready to run when the local time is within the run window for the queued submission.
 3. Issue the **run** action to run the queued submission using the POST `/zosmf/mgmt-services/service-instances/{submission-id}/actions/run` API. The submission-id is included in the response from the Create new submission API. The submission only runs if the status is **Ready to run** and the local time is within the run window for the submission (between **siRunAfter** and **siExpires**).

Delete a service submission

Use this operation to delete a service submission from z/OS Management Services Catalog (zMSC).

HTTP method and URI path

```
DELETE /zosmf/mgmt-services/rest/service-instances/{object-id}
```

In this request:

{object-id}

Identifies the service submission to be deleted from z/OS Management Services Catalog.

Query parameters

None.

Description

This operation deletes an existing service submission.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request resulted in the deletion of the service submission.

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either have the zMSC Administrator role or be the creator of the service submission.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 204 (No Content) is returned.

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

None.

Example HTTP interaction

The example [Figure 446 on page 990](#) shows a request to delete a service submission from z/OS Management Services Catalog.

```
DELETE /zosmf/mgmt-services/rest/service-instances/<object-id>
```

Figure 446. Sample request to delete a service submission from the z/OS Management Services Catalog

Get the details of a service submission

Use this operation to get details of a specific service submission in z/OS Management Services (zMSC).

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/service-instances/{object-id}
```

In this request:

{object-id}

Identifies the service submission to be retrieved from the z/OS Management Services Catalog.

Query parameters

None.

Description

This operation retrieves the details of a specific service submission.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request resulted in a domain retrieval. A response body is provided as, described in [Table 522 on page 991](#).

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either be authorized as a zMSC User or zMSC Administrator.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned, and a response body is provided, as described in [Table 522 on page 991](#).

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service submission returns a response body that contains a JSON object with details about the service submission.

Table 522. Fields for a get service submission object			
Field	Type	Guaranteed/ Optional	Description
siName	String	Guaranteed	Name of the service submission. This is the same as the name of catalog service that the service submission was created from.
objectId	String	Optional	The unique object ID of the service submission.
objectUri	String	Guaranteed	The URI of the service submission.
objectUrl	String	Guaranteed	The URL of the service submission.
siCreatedBy	String	Guaranteed	The user who created the service submission.
siCreatedTime	String	Guaranteed	When the service submission was first created.
siLastModifiedTime	String	Guaranteed	When the service submission was last modified. If no modifications have occurred, this is the time that the service submission was created.
siLastModifiedBy	String	Guaranteed	The last user to modify this service.
siQueuedTime	String	Optional	When the service submission entered the queued status.
siRunTime	String	Optional	When the service submission entered the running status.

Table 522. Fields for a get service submission object (continued)

Field	Type	Guaranteed/ Optional	Description
siStatus	String	Guaranteed	Status of the service submission Status include: <ul style="list-style-type: none"> • Queued • Running • Failed • Completed • Terminated • Draft
siTargetSystem	String	Guaranteed	System that the service submission is running on.
siRunAutomatically	Boolean	Guaranteed	If true, the service submission can run automatically. If false, the service submission must be run manually.
siWorkflowInstanceName	String	Optional	The workflow name in the workflows task that is created to run the service submission. This is seen whether the service submission has started.
siWorkflowInstanceKey	String	Optional	If the service submission has started, the workflow key in the workflows task that is created to run the service submission.
siCatalogServiceId	String	Guaranteed	The UUID of the category service that the service submission was created from.
siCatalogServiceDeleted	Boolean	Guaranteed	Indicates whether the underlying catalog service submission is deleted.
siRunAfter	String	Optional	The service submission cannot be run before this time if it is returned.
siExpires	String	Optional	The service submission cannot be run after this time if it is returned.
siChangeRecord	String	Optional	User entered value. Intended to be associated with an external change management system.
siActivityTimeline	JSONArray of ActivityTimeline Objects	Guaranteed	List of objects that display the history of events that pertain to this service submission. For more information, see Table 523 on page 993 .
siInputs	JSONObject	Optional	JSONObject with a key value pair of the variable ID and its value.
siPreviousInputs	JSONObject	Optional	JSONObject with a key value pair of variable values and the value it was set to before its current value.
mgmtsvcsVersion	String	Guaranteed	Identifies the management services version of the catalog service.

Table 522. Fields for a get service submission object (continued)

Field	Type	Guaranteed/Optional	Description
mgmtsvcsVersionSupported	Boolean	Guaranteed	Indicates whether GET, POST, PUT, and DELETE operations are allowed for the catalog service. Operations are not allowed if the version associated with the level of the z/OS Management Services plug-in is LESS than that of this object. <ul style="list-style-type: none"> • true if operations are allowed. • false if operations are not allowed
siCatalogServiceDefinition	JsonObject	Optional	JSON Object that contains the service submission's catalog service definition. If the catalog service submission gets deleted, this property is not returned and siCatalogServiceDeleted is set to true. For more information, see Table 505 on page 963 .
siBeingModifiedBy	String	Guaranteed	The user who is modifying the service submission. Other users must use the Acquire lock action to be able to update the service submission.
siJobStatement	String	Guaranteed	The JOB statement that is sent to the workflow engine when the service submission is run. If the string is empty, the workflow engine's default JOB statement is used.
siTestRun	Boolean	Guaranteed	True if the service submission is a test run. False if it is a standard run.

Table 523. Fields for an ActivityTimeLine object

Field	Type	Guaranteed/Optional	Description
actionName	String	Guaranteed	Name of the action that is performed.
actionComment	String	Optional	Comment provided by the user that is performing the action.
actionTime	String	Guaranteed	When the action was performed.
actionUser	String	Guaranteed	The user who performed the action.
actionAdditionalInfo	Object	Guaranteed	Object that contains any potential additional information about the action.

Example HTTP interaction

The example in [Figure 447 on page 994](#) shows a request to get the details of a service submission in z/OS Management Services.

```
GET /zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab
```

Figure 447. Sample request to get the details of a specific service submission in the z/OS Management Services.

The following example shows a response body that contains the details of a service submission.

```
{
  "siName": "Great New Service",
  "objectId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "objectUri": "/zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/service-instances/aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaab",
  "siCreatedBy": "ConnerF",
  "siCreatedTime": "2019-12-10T09:10:43.221Z",
  "siLastModifiedTime": "2019-12-20T09:10:43.221Z",
  "siLastModifiedBy": "ConnerF",
  "siCatalogServiceId": "aaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaabb",
  "siCatalogServiceDeleted": false,
  "siQueuedTime": "2019-12-20T09:10:40.221Z",
  "siStatus": "Queued",
  "siTargetSystem": "PLEX1.TEST",
  "siRunAutomatically": false,
  "siRunAfter": "2019-12-22T09:12:48.291Z",
  "siExpires": "2019-12-24T09:12:48.291Z",
  "siChangeRecord": "A180418",
  "siBeingModifiedBy": "VC",
  "siJobStatement": "",
  "siTestRun": false,
  "siActivityTimeline": [
    {
      "actionName": "SUBMIT",
      "actionComment": null,
      "actionTime": "2020-07-09T20:18:54.441Z",
      "actionUser": "zosmfad",
      "actionAdditionalInformation": null
    }
  ],
  "siInputs": {
    "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "SMS Managed",
    "baaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "SMS252",
    "daaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "Some variable value",
    "eaaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": null
  },
  "siPreviousInputs": {
    "caaaaaaa-aaaaaaaa-aaaaaaaa-aaaaaaaa": "NonSMSManaged"
  },
  "siCatalogServiceDefinition": {
    "csName": "Great New Service",
    "csDescription": "A description of this great new services.",
    "objectId": "128c778f-94444030-b2d13509-87640705",
    "objectUri": "/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "objectUrl": "https://mvshost1.ibm.com/zosmf/mgmt-services/rest/catalog-services/128c778f-94444030-b2d13509-87640705",
    "csCategory": "Catalog Management",
    "csState": "published",
    "createdTime": "2019-11-15T19:12:45.223Z",
    "csTags": [ "tag1", "tag2" ],
    "csDefaultInstanceExpiryInterval": "PT98H",
    "csCreatedBy": "suser1",
    "csLastModifiedTime": "2019-12-22T09:12:48.291Z",
    "csLastModifiedBy": "suser2",
    "csOriginalWfDefinitionPath": "/u/suser1/workflows/createZosThing.xml",
    "csOriginalWfVariablePropertiesPath": "/u/suser1/workflows/wfVariable.properties",
    "mgmtsvcsVersion": 0100,
    "mgmtsvcsVersionSupported": 0100,
    "csExpiryUpdateableInRunner": "false",
    "csPages": [
      {
        "csPageName": "page 1",
        "csPageDescription": "Detailed information about this page",
        "csInputDefinitions": {
          "createZosThingWfVar1": {
            "csInputLabel": "The first required input",
            "csInputDescription": "Detailed information about this input."
          }
        }
      }
    ]
  }
}
```



```

        "csInputType" : "string",
        "csInputWidget" : "checkbox",
        "csInputValues" : ["value1", "value2", "value3", "value4"],
        "csInputWfVarName" : "createZosThingWfVar1",
        "csInputValidation" : {
            "minLength" : "20",
            "maxLength" : "60"
        }
    },
    "createZosThingWfVar2": {
        "csInputLabel" : "The second input",
        "csInputDescription" : "This input is required if Input
1 has the value DATASET.",
        "csInputType" : "integer",
        "csInputWidget" : "radio-button",
        "csInputWfVarName" : "createZosThingWfVar2",
        "csInputValidation" : {
            "minLength" : "20",
            "maxLength" : "60"
        }
    },
    "createZosThingWfVar2a": {
        "csInputLabel" : "Depends on Input 2",
        "csInputDescription" : "This input only available when
Input 2 is false, and then it is optional.",
        "csInputType" : "boolean",
        "csInputWidget" : "radio-button",
        "csDefaultValue" : true,
        "csInputWfVarName" : "createZosThingWfVar2a"
    }
}
"csInputOrder": ["createZosThingWfVar1", "createZosThingWfVar2a"]
}]
"csInputLogic": [
    {
        "csControlInputDefinitionId": "createZosThingWfVar2a",
        "csControlVariableLogiccsControlVariableComparator": "equal",
        "csControlVariableComparatorValue": "DATASET",
        "csDependentVariableId": "Input 2",
        "csDependentVariableRequired": true
    },
    {
        "csControlInputDefinitionId": "createZosThingWfVar1",
        "csControlVariableComparator": "equal",
        "csControlVariableComparatorValue": false,
        "csDependentVariableId": "Input 2a",
        "csDependentVariableRequired": false
    }
]
}
"mgmtsvcsVersion": 100,
"mgmtsvcsVersionSupported": true
}

```

Modify an existing service submission

Use this operation to modify properties of a specific queued service submission in z/OS Management Services Catalog (zMSC).

HTTP method and URI path

```
PATCH /zosmf/mgmt-services/rest/service-instances/{object-id}
```

In this request:

{object-id}

Identifies the service submission to modify in z/OS Management Services Catalog.

Query parameters

None.

Description

This operation changes a specific queued service submission from z/OS Management Services Catalog.

On successful completion, HTTP status code 204 (No Content) is returned, which indicates that the request successfully modified the specified service submission.

Request content

A request body that identifies the changed elements that are required for this request. Changed properties are replaced in their entirety. All properties are optional, although at least one must be present for a successful operation.

Table 524. Fields for modifying a service submission object			
Fields	Type	Required/Optional	Description
siInputs	JSONObject	Optional	JSONObject with a key value pair of the variable ID and its value.
siRunAfter	String	Optional	The service submission cannot be run before this time.
siExpires	String	Optional	The service submission cannot be run after this time.
siTargetSystem	String	Optional	The system that the service submission is run on.
siChangeRecord	String	Optional	User entered value. Intended to be associated with an external change management system.
siJobStatement	String	Optional	The JOB statement that is sent to the workflow engine when the service submission is run. If it is not provided, the workflow engine's default JOB statement is used.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either have the zMSC Administrator role or be the creator of the service submission.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 204 (No Response) is returned.

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

None.

Example HTTP interaction

Figure 448 on page 997 shows an example of a request to update a service submission from the z/OS Management Services Catalog.

```
PATCH /zosmf/mgmt-services/rest/service-instances/f684f8d9-62dd-4553-ac7b-054627fff188
{
  "siInputs": { "977a04ee-5d96-4fe1-ad4c-a40ec5616325": "SMS253",
                "fa5bb8e2-859c-4776-a206-3d6221a18625": "Some variable value2"},
  "siTargetSystem": "PLEX3.SYS1"
}
```

Figure 448. Sample request to update a service submission from the z/OS Management Services Catalog

Perform an action on a service submission

Use this operation to perform an action on a service submission from z/OS Management Services Catalog.

HTTP method and URI path

```
POST /zosmf/mgmt-services/rest/service-instances/{object-id}/actions/{action}
```

In this request:

{object-id}

Identifies the service submission to perform an action on.

{action}

Identifies which action is taken on the service submission.

Valid actions are:

- submit
- run
- terminate
- copy
- comment

Query parameters

None.

Description

This operation performs the specified action on an existing service submission.

On successful completion, HTTP status code 204 (No Content) is returned, which indicates that the request successfully completed the specified action on the service submission.

Action Descriptions

submit

Submit can only be performed on a service submission that is in the Draft State. If the service submission has values for all of its required fields, the submit action succeeds. It then moves the service submission into either a running or queued state, depending on if the service submission was set to run automatically or manually.

Required fields for a service submission depend on the service and the user settings.

These fields are not being provided as part of the submit action. They must already be set in the service submission by a POST or PATCH request.

Table 525. Fields that may be required to be provided before a draft service submission can be successfully submitted

Field	Type	Required When	Description
siTargetSystem	String	Always required	System that the service submission is run on.
siChangeRecord	String	Depends on the Catalog Settings	String that allows the user to associate the new service submission with a change record identifier from an external change management system.
siInputs	JSONObject	Depends on the service definition	JSONObject with a key value pair of the variable ID and its value.
siRunAfter	String	Depends on the service definition	The service submission cannot be run before the specified time.
siExpires	String	Depends on the service definition	The service submission cannot be run after the specified time.

run

Run a queued service submission. This moves the service submission into the running state, and then creates and starts the automation for the workflow instance.

terminate

Cancel a running service submission. This stops the workflow automation and moves the service submission into a canceled state.

copy

The "copy" action creates a new draft of the service submission, which keeps all of the original service submission's inputs and run properties.

comment

Add a comment to the service submission's activity timeline.

Request content

Table 526. Fields for adding a comment to a service submission object

Field	Type	Required/Optional	Description
siComment	String	Required if action is "comment." Optional for all other actions	The comment to be added to the activity log of the service submission. Limited to 300 characters.

Otherwise, no content is expected.

Authorization requirements

The user's z/OS user ID must have access to z/OSMF and either have the zMSC Administrator role, or be the creator of the service submission in order to perform "Run", "Cancel", or "Submit" actions. Any user with access to zMSC can perform the "Comment" action.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 204 (No Content) is returned.

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

None.

Example HTTP interaction

The example in [Figure 449 on page 999](#) shows a request to run a service submission from z/OS Management Services Catalog.

```
POST /zosmf/mgmt-services/rest/service-instances/f684f8d9-62dd-4553-ac7b-054627fff188/actions/run
```

Figure 449. Sample request to run a service submission

Get the list of JCL JOB statements

This operation retrieves the list of JCL JOB statements that are defined in the z/OS Management Services Catalog global settings.

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/settings/job-statements
```

Query parameters

None.

Description

This operation retrieves the list of JCL JOB statements that are defined in the z/OS Management Services Catalog global settings.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request returned the list of JCL JOB statements.

Request content

None.

Authorization requirements

The user's z/OS user ID must have access to the z/OS Management Services Catalog task.

For more information about authorization requirements for z/OS Management Services Catalog roles, see [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned, and the response body is provided. See [“Response content” on page 1000](#).

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service returns a response body, which contains a JSON object that is described in the following tables. See [“Response content” on page 1000](#).

Table 527. Response from a get list of job statements request		
Field	Type	Description
lockHolder	String	The user who currently holds the lock on the job statement resource.
jobStatements	Array of job statement items	Array of job statement items, which are defined below.
settingsLastModifiedBy	String	The user ID of the user that last modified the JOB statements list.
settingsLastModifiedTime	String	The ISO-8601 formatted UTC time when the job statements list was modified.

Example HTTP interaction

In Figure 450 on page 1000, a request is submitted to get the list of job statements that are defined in the z/OS Management Services Catalog global settings.

```
GET /zosmf/mgmt-services/rest/settings/job-statements
```

Figure 450. Sample request to get the list of job statements that are defined in the z/OS Management Services Catalog global settings

The following example is the response body for the example.

```
{
  "lockHolder": "zosmfad",
  "settingsLastModifiedBy": "",
  "settingsLastModifiedTime": "2021-02-15T21:12:59.472Z",
  "jobStatements": [
    {
      "name": "cardClassA",
      "jobStatement": "//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
        //MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER"
    },
    {
      "name": "cardClassB",
      "jobStatement": "//IZUWFJB JOB (ACCTINFO),CLASS=B,MSGCLASS=0,
        //MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER",
      "description": "Use for jobs that..."
    }
  ]
}
```

Get the list of target systems

This operation retrieves the list of target systems that are defined in the z/OS Management Services Catalog (zMSC) global settings.

HTTP method and URI path

```
GET /zosmf/mgmt-services/rest/<version>/settings/target-systems
```

Query parameters

None.

Description

This operation retrieves the list of target systems that are defined in the z/OS Management Services Catalog global settings.

On successful completion, HTTP status code 200 (OK) is returned, which indicates that the request returned the list of target systems.

Request content

None.

Authorization requirements

The user's z/OS user ID must have user access to the z/OS Management Services Catalog task.

For more information, see [“z/OS Management Services Catalog services” on page 953](#).

Status codes

On successful completion, HTTP status code 200 (OK) is returned and a response body is provided, as described in [“Response content” on page 1001](#).

For more information about status codes, see [“HTTP status codes” on page 955](#).

Response content

On successful completion, the service returns a response body that contains a JSON Object that is described in the following table. See [Table 528 on page 1001](#).

Table 528. Response from a get list of target systems request		
Field	Type	Description
lockHolder	String	The user ID of the user that is modifying the target systems settings.
targetSystems	Array of system objects	The list of target systems that are available for the plug-in to use.
settingsLastModifiedBy	String	The user ID of the user that last modified the target system list.
settingsLastModifiedTime	String	The ISO-8601 formatted UTC time when the target system list was modified.

Example HTTP interaction

The Figure 451 on page 1002 example shows a request to get the list of systems in the z/OS Management Services and example response body that contains two system entries.

```
GET /zosmf/mgmt-services/rest/settings/target-systems
```

Figure 451. Sample request to get the list of target systems

The following example shows the response body to the Figure 451 on page 1002.

```
{
  "lockHolder": "",
  "targetSystems": [
    {
      "systemNickName": "SY1",
      "systemName": "PLEX1.SY1",
      "changeRecordRequired": false,
      "enabled": true,
      "additionalInformation": null,
      "deleted": false,
      "systemLastModifiedTime": "",
      "systemLastModifiedBy": ""
    }
  ],
  "settingsLastModifiedBy": "ibmuser",
  "settingsLastModifiedTime": "2021-11-23T16:53:44.937Z"
}
```

z/OSMF information retrieval service

The z/OSMF information retrieval service is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. This service allows the caller to query the version and other details about the instance of z/OSMF running on a particular system.

z/OSMF information includes the following details:

- SAF realm
- z/OSMF listening port
- z/OSMF version and release
- Installed plug-ins and plug-in build levels
- Indicates the z/OS operating system level.

With this information, a calling program can determine which z/OSMF plug-ins and API functions are available for use on a given system. For information, see [“Retrieve z/OSMF information” on page 1003](#).

Using the Swagger interface

You can use the Swagger interface to display information about the z/OSMF information retrieval service REST API. For more information, see [“Using the Swagger interface” on page 1](#).

Required authorizations

None.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate hypertext transfer protocol (HTTP) status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the information retrieval service.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the information retrieval service are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Retrieve z/OSMF information

You can use this operation to retrieve information about z/OSMF on a particular z/OS system.

HTTP method and URI path

```
GET /zosmf/info
```

where **zosmf/info** identifies the z/OSMF information retrieval service.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data. The following JSON object is received as output from the request.

```
{
  "zosmf_saf_realm": "SAF-profile-prefix",
  "zosmf_port": "zosmf-server-port-number",
  "zosmf_full_version": "zosmf-release-level",
  "plugins": [
    {
      "pluginVersion": "plugin-fmid-build-level",
      "pluginStatus": "plugin-status",
      "pluginDefaultName": "plugin-name"
    },
    :
    {
      "pluginVersion": "plugin-fmid-build-level",
      "pluginStatus": "plugin-status",
      "pluginDefaultName": "plugin-name"
    }
  ],
  "api_version": "api-version",
  "zos_version": "zos-release",
  "zosmf_version": "zosmf-version",
  "zosmf_hostname": "host-system-URL"
}
```

where:

zosmf_saf_realm

Realm associated with the system on which z/OSMF is installed. Usually, this is the sysplex name.

zosmf_port

Port number for SSL encrypted traffic for the active instance of z/OSMF on the z/OS system.

zosmf_full_version

Indicates the z/OSMF version, further qualified by a service level.

plugins

Array of zero, one, or more elements that contain information about each of the installed z/OSMF plug-ins. If no plug-ins are installed, this area is empty.

Each element contains the following attributes:

pluginVersion

Indicates the plug-in version (FMID) and build level.

pluginStatus

Indicates the status of the plug-in. The status is reported for IBM-supplied plug-ins only. For an external application, the status is blank.

The following values are valid:

ACTIVE

The plug-in is running.

INSTALLED

The plug-in is installed, but not running.

UNINSTALLED

The plug-in was installed in a previous z/OSMF configuration, but is not installed in the current configuration. This status can result when a plug-in is removed from z/OSMF.

After a plug-in is started, its status remains as ACTIVE, even if the plug-in is later stopped.

pluginDefaultName

Indicates the plug-in name.

api_version

Version of the z/OSMF information retrieval service and the JSON object structure used for this request. The version sequence starts at 1, and is incremented if the service or the JSON structure changes.

zos_version

Indicates the z/OS operating system level. The following values are valid:

04.24.00

Indicates that the z/OS level is V2R1.

04.25.00

Indicates that the z/OS level is V2R2.

04.26.00

Indicates that the z/OS level is V2R3.

zosmf_version

Indicates the z/OSMF level. The following values are valid:

24

Indicates that the z/OSMF level is V2R1.

25

Indicates that the z/OSMF level is V2R2.

26

Indicates that the z/OSMF level is V2R3.

zosmf_hostname

Indicates the hostname or IP address of the z/OS system on which z/OSMF is installed

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

None.

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 1002](#).

The response also includes a JSON object that contains the retrieved data. For details, see [“Content type used for HTTP response data” on page 1003](#).

Example request

In the following example, the GET method is used to retrieve information about z/OSMF.

```
GET /zosmf/info HTTP/1.1
Host: host.name.com
```

Figure 452. Sample request to retrieve z/OSMF information

Example response

For a successful request, the HTTP response includes a JSON document containing the requested information.

```

HTTP/1.1 200 OK
Date: Wed, 06 Mar 2013 06:39:28 +0000GMT
Content-Type: text/plain
Connection: close

{
  "zosmf_saf_realm": "SAFRealm",
  "zosmf_port": "443",
  "zosmf_full_version": "24.02",
  "plugins": [
    {
      "pluginVersion": "hsma210.spe2;driver05;2014-02-11T03:21:53",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Import Manager",
      "pluginVersion": "hsma213.spe2;driver05;2014-02-11T10:17:27",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "WorkloadManagement",
      "pluginVersion": "H0X7790;driver122;2014-02-18T00:00:00Z",
      "pluginDefaultName": "IBM SDSF",
      "pluginVersion": "hsma216;DRIVER04;2014-01-14T19:03:40",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Capacity Provisioning",
      "pluginVersion": "hsma214.spe2;driver05;2014-02-11T08:28:24",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Software Deployment",
      "pluginVersion": "hsma21a;pm93903;2013-08-12T03:52:53",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "NetworkConfigurationAssistant",
      "pluginVersion": "hsma215.spe2;driver05;2014-02-11T10:16:30",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "IncidentLog",
      "pluginVersion": "hsma211.spe2;driver05;2014-02-11T03:29:52",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "ISPF",
      "pluginVersion": "hsma212;DRIVER4B;2014-01-14T12:43:43",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "ResourceMonitoring",
      "pluginVersion": "hsma217.spe2;driver05;2014-02-11T12:01:40",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Workflow"
    }
  ],
  "api_version": "1",
  "zos_version": "04.24.00",
  "zosmf_version": "24",
  "zosmf_hostname": "host.name.com"
}

```

z/OSMF settings services

The z/OSMF settings services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with the z/OSMF settings for z/OSMF plug-ins and the z/OSMF server. Depending on the changes specified, the changes can be scoped to individual users, plug-ins, or the z/OSMF server.

Table 529 on page 1006 lists the operations that the z/OSMF settings services provide.

Table 529. Operations provided through the z/OSMF settings services	
Operation	HTTP method and URI path
Put	PUT /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	PUT /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
Get	GET /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	GET /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
Delete	DELETE /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	DELETE /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>

Required authorizations

The user must be logged in to z/OSMF, and must have READ access to the SAF profile that was registered for the plug-in and task making the request.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all z/OSMF settings services as input and output for the requested operations. The attributes that are provided in the JSON object depend on the requested operation.

```
{
  "value": "data-value",
  "version": "structure-version",
  "messages": "z/OSMF-messages",
  "update": true|false
}
```

Where:

data-value

The value that is added, updated, retrieved, or removed by the z/OSMF settings services. Any data type is supported including JSON objects, JSON arrays, and scalars. The value is required.

structure-version

Version of the z/OSMF settings services and the JSON object structure that is used for this request. The version sequence starts at 1.0.0, and is incremented only if the services or the JSON structure changes. The version the client supports is required as input to the request. The z/OSMF settings services are backward compatible for $n-2$ versions, and accepts requests for each version it supports. If the version specified by the client is not supported or if no version is specified, the service returns an error message.

z/OSMF-messages

z/OSMF messages received during the request. The *messages* attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

update

An optional input attribute, which indicates that the service is updating or replacing an existing JSON object. If you set the value to *true*, the service updates the key-value pairs you specified for the *value* attribute and preserves any other data persisted in the JSON object. You can set this attribute to *true* only when the data type is a JSON object or JSON array. If you omit this attribute or set it to *false*, the service deletes the existing JSON object and creates a new JSON object that contains only the key-value pairs you specified for the *value* attribute.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

HTTP 200 OK

Success.

HTTP 400 Bad request

Request contained incorrect parameters.

HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the z/OSMF settings services.

HTTP 404 Bad URL

Target of the request (a URL) was not found.

HTTP 500 Internal server error

Programming error.

Error logging

Errors from the z/OSMF settings services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Persist z/OSMF settings for a user or plug-in

You can use this operation to persist z/OSMF settings for a user, z/OSMF plug-in, or the z/OSMF server.

HTTP method and URI path

```
PUT /zosmf/settings/system/<resourcePath>?version=1.0.0
PUT /zosmf/settings/user/system/<resourcePath>?version=1.0.0

PUT /zosmf/settings/zosmf/<resource-path>?version=1.0.0
PUT /zosmf/settings/user/zosmf/<resource-path>?version=1.0.0

PUT /zosmf/settings/app/<plugin-id>/<task-id>/<resource-path>?version=1.0.0
PUT /zosmf/settings/user/<plugin-id>/<task-id>/<resource-path>?version=1.0.0
```

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

Query parameters

None.

Standard headers

Use the following standard HTTP headers with this request:

```
Accept: application/json
Content-Type: application/json
```

Custom headers

None.

Request content

Your request must include a JSON object that contains the value to be persisted and the version. For more details, see [“Content type used for HTTP request and response data”](#) on page 400.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 400](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more details, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the current data after being modified. For more details, see [“Content type used for HTTP request and response data” on page 715](#).

Example

In this example, assume that you want to enable the z/OSMF services "z/OS jobs REST interface" and "z/OS data set and file REST interface." Also, you want to disable the Sysplex Management plug-in. To persist data that satisfies this criteria, submit the request that is depicted in [Figure 453 on page 1009](#):

```
PUT /zosmf/settings/zosmf/services?version=1.0.0
Host: zosmf1.yourco.com
Accept: application/json
Content-Type: application/json
{
  "value": {
    "IZU_REST_FILE": "Y",
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_JOB": "Y"
  }
}
```

Figure 453. Sample request to persist user-specific z/OSMF settings data

Retrieve z/OSMF settings data

You can use this operation to retrieve z/OSMF settings data that is persisted for a specific user or application.

HTTP method and URI path

```
GET /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
GET /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
```

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash

(/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Required authorizations

See [“Required authorizations” on page 1006](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more information, see [“Error handling” on page 717](#).

The response also includes a JSON object that contains the retrieved data. For more information, see [“Content type used for HTTP request and response data” on page 715](#).

Example

In this example, assume that you want to query the enablement status of the z/OSMF services and plug-ins on your system. To retrieve the status, submit a request like the one that is shown in [Figure 454 on page 1010](#):

```
GET: /zosmf/settings/zosmf/services?version=1.0.0  
Host: zosmf1.yourco.com
```

Figure 454. Sample request to retrieve z/OSMF settings data

A sample response is shown in [Figure 455 on page 1011](#). In the example, the enablement status is returned for the following z/OSMF services:

- Sysplex Management plug-in is disabled.
- z/OS jobs REST interface is enabled.
- z/OS data set and file REST interface is enabled.


```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close

{
  "version": "1.0.0",
  "value": {
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_FILE": "Y",
    "IZU_REST_JOB": "Y"
  }
}
```

Figure 455. Sample response from a request to retrieve z/OSMF settings data

Delete z/OSMF settings data

You can use this operation to delete z/OSMF settings data that is persisted for a specific user or application.

HTTP method and URI path

```
DELETE /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
DELETE /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
```

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

Custom headers

None.

Request content

None.

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Required authorizations

See [“Required authorizations” on page 1006](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information, see [“Error handling” on page 717](#).

The response also includes the updated JSON object. For more information, see [“Content type used for HTTP request and response data” on page 715](#).

Example

In this example, assume that you want to delete the persisted data for the z/OS jobs REST interface. These z/OSMF services are represented by the IZU_REST_JOB key under the /services portion of the request URI. To delete the persisted data for this service, submit a request like the one shown in [Figure 456 on page 1012](#).

```
DELETE /zosmf/settings/zosmf/services/IZU_REST_JOB?version=1.0.0
Host: zosmf1.yourco.com
```

Figure 456. Sample request to delete persisted z/OSMF settings data

A sample response is shown in [“Delete z/OSMF settings data” on page 1011](#). On completion, the delete request returns the value of "IZU_REST_JOB" key, which is now null.

```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close
{
  "version": "1.0.0",
  "value": null
}
```

Figure 457. Sample response from a request to retrieve z/OSMF settings data

Following this deletion, assume that you were to query the persisted data for the z/OSMF services, as shown in [“Retrieve z/OSMF settings data” on page 1009](#).

```
GET: /zosmf/settings/zosmf/services?version=1.0.0
Host: zosmf1.yourco.com
```

Figure 458. Sample request to retrieve z/OSMF settings data

The response body no longer contains data for the z/OS jobs REST interface.

```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close
{
  "version": "1.0.0",
  "value": {
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_FILE": "Y"
  }
}
```

Figure 459. Sample response from a request to retrieve z/OSMF settings data

z/OSMF authentication services

The z/OSMF authentication services API is provided for z/OSMF tasks and vendor applications. This API is used to obtain or delete authentication tokens (a JSON Web Token and an LTPA token) on the user's authentication request when logging in to or out of z/OSMF. This API can also be used to change a z/OSMF user's password.

Table 530. z/OSMF authentication services method

Operation	HTTP method and URI path
“Log in to the z/OSMF server” on page 1014	POST /zosmf/services/authenticate
“Change the user password or passphrase” on page 1016	PUT /zosmf/services/authenticate
“Log out of the z/OSMF server” on page 1019	DELETE /zosmf/services/authenticate

For information about enabling the z/OSMF server to produce JSON Web Tokens, see [Enabling JSON Web Token support](#) in *IBM z/OS Management Facility Configuration Guide*.

Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

HTTP 200 OK

Request was processed successfully.

HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

HTTP 503 Service unavailable

Server is not available.

Error logging

Errors from the z/OSMF authentication services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Log in to the z/OSMF server

You can use the POST method to log in to the z/OSMF server and obtain authentication tokens. This service creates a JSON Web Token, an LTPA token, or both, and returns the tokens to the requester.

HTTP method and URI path

```
POST /zosmf/services/authenticate
```

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

Query parameters

None.

Standard headers

Use the following standard HTTP headers with this request:

Content-Type: application/x-www-form-urlencoded

Indicates that the body of the HTTP message is a query string, consisting of name=value pairs, with each pair separated by a single ampersand (&).

Authorization: Basic <credentials>

In the context of an HTTP transaction, basic access authentication is a method for an HTTP user agent (such as a web browser) to provide a user ID and password with a request. In basic HTTP authentication, a request contains a header field in the form of **Authorization: Basic <credentials>**, where **<credentials>** is the Base64 encoding of ID and password, joined by a single colon (:).

Customer headers

X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or an empty string (" "). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

Request body

None.

Response Body

Table 531. Response body for a "log in to the z/OSMF server" request	
Field name	Description
returnCode	Identifies the category of errors
reasonCode	Specified file is either not found or cannot be opened.
message	Describes the text information of the login result.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 4.

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error occurred.

Table 532. HTTP error response codes for a "log in to the z/OSMF server" request				
HTTP error status code	Return code	Reason code	Message	Description
200	0	0	Success.	User logs in to z/OSMF server successfully.
401	8	1	Login failed. Check whether the user ID and password you use for the Basic Auth is correct, and if the user ID has the required SAF permissions.	Check whether the user ID and password you use for the Basic Auth is correct, and if the user ID has the required SAF permissions.
401	8	12	The password or passphrase is expired.	Your password or passphrase is expired. Reset the password or passphrase.
401	8	28	The user ID was revoked.	Your user ID was revoked. Contact your system administrator.
401	4	1	Login failed. The Basic Auth information in the request header is incorrect.	Correct any errors in the Basic Auth information, which is in the request header.
401	4	40	The request failed because an internal error occurred.	The request failed because an internal error occurred. Contact your system administrator.

Note: In the **General Settings > Home Page**, if the option **Display error details when login fails** is selected, the error message indicates when a password is expired (return code 8, reason code 12), or a user ID was revoked (return code 8, reason code 28). Otherwise, if the option **Display error details when login fails** is not selected, an expiration or revocation error returns only the general failure message: return code 8, reason code 1.

Example

In the following example, the POST method is used to log in to the z/OSMF server and obtain a JSON Web Token for the user. On completion, the token is saved as a web cookie.

```

POST /zosmf/services/authenticate HTTP/1.1
Host: your.company.com
Referer: https://your.company.com/zosmf
Content-Type: application/x-www-form-urlencoded
cookie: "jwtToken=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2t1b190eXB1IjoiaWVhcmVhIiwic3ViIjoiem9zbWZhZCIsInVwbiI6Inpvc21mYWQlLCJncm91cHMlOi01siQ0VBR1AiLCJDUe9DVFJMIiwic1BPVVVfU1kiLCJJWlVBRE1JTtIsIldMTUdSUCIsIkN0ZHI10sInJlYWxtIjoiaU0FGUmVhbG0iLCJpc3MiOiJ6T1NNRiIsImV4cCI6MTU0ODg2MTM0NiwiawF0IjoxtQ40DMxOTQ2fQ.Bfc9MqPSRfn-rz0Gryf_24klKD8iqZB0TWwYwYB9osdKpFDbS2wKtqKMyBx6gcPX649Uk9mhSJ1VEAz71A0gOLi9kA28rMj1mQZbimkzNzXEPpP37HG5Ve8aGvtxCdsyPKUQAQC9DRCD-B-MGLjwCaLokRI7BjtDhriwK17yBVAx0GV7Gs8arFQzsTZSLRj3VWJ-wpu0JATRUCsjBLPE7inZxxzGIlh0eyM0K_0EH6YB_RTEsdF21SbNy36dA2aJaaYHLw9j1-fJ1VNwMf4ipFy_x2em8bdLYIhVT0ujLttjWmd2M9dEnYGs_rt6xpKfBCILnN0yXowrrwEchEPE1Q"
cache-control: no-cache
response body:
{"returnCode":0,"reasonCode":0,"message":"Success."}

```

Figure 460. Log in and obtain a JSON Web Token for user authentication

Change the user password or passphrase

You can use a PUT request to change a user password or password phrase (passphrase). This service is available when you install the PTF for APAR PH34912.

HTTP method and URI path

```
PUT /zosmf/services/authenticate
```

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

Query parameters

None.

Standard headers

Use the following standard HTTP header with this request:

```
Content-Type: application/json
```

Customer headers

X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or the empty string (" "). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

Request content

The request content is expected to contain a JSON object. See [Table 533 on page 1016](#).

Table 533. Request content for a change user password or passphrase request			
Field name	Type	Required or optional	Description
userID	String	Required	User ID.

Table 533. Request content for a change user password or passphrase request (continued)

Field name	Type	Required or optional	Description
oldPwd	String	Required	The current (or old) password or passphrase for the user ID.
newPwd	String	Required	The new password or passphrase for the user ID. Ensure that the same type of authentication value is used: Replace a password with a password; replace a passphrase with a passphrase.

Response Body

Table 534. Response body for a change user password or passphrase request

Field name	Description
returnCode	Identifies the category of errors
reasonCode	Identifies the specific errors.
message	Describes the text information of the change request.

Required authorizations

See [“Required authorizations” on page 818.](#)

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4.](#)

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error occurred.

Table 535. HTTP error response codes for a change user password or passphrase request

HTTP error status code	Return code	Reason code	Message	Description
200	0	0	Success.	The password or passphrase is changed.
400	4	4	The user ID is not defined to RACF.	The user ID is not defined to RACF. Verify that the user ID is spelled correctly. This value is case-sensitive.
400	4	21	The user ID is not valid. It is either empty or exceeds the character limit.	The user ID is not valid. It is either empty or exceeds the character limit.
400	4	22	The current (old) password cannot be null.	The current (old) password cannot be null.
400	4	23	The new password cannot be null.	The new password cannot be null.

Table 535. HTTP error response codes for a change user password or passphrase request (continued)				
HTTP error status code	Return code	Reason code	Message	Description
500	8	2	Change password failed. Check whether the user ID and oldPwd you provide is correct.	Determine whether the user ID and old password you provided are correct. You cannot change a password to a passphrase, or vice versa.
500	8	16	Change password failed. newPwd is invalid.	At least one of the following conditions is true: <ul style="list-style-type: none"> • The new password or passphrase is not valid. • A new password was specified with a current password, or a new passphrase was specified with a current passphrase. • A new passphrase was specified with a PassTicket as the current password, but the user does not currently have a passphrase. • A password or passphrase change is disallowed now because the minimum password-change interval has not elapsed.
500	8	28	The user ID was revoked.	Your user ID was revoked. Contact your system administrator.
500	8	29	The request body is not in JSON format.	The request body is required to be in JSON format. Convert the request body to JSON format.
500	4	40	The request failed because an internal error occurred.	The request failed because an internal error occurred. Contact your system administrator.

Note: In the **General Settings > Home Page**, if the option **Display error details when login fails** is selected, the error message indicates when a user ID was revoked (return code 8, reason code 28) or a user ID profile is not defined to RACF (return code 4, reason code 4). Otherwise, if the option **Display error details when login fails** is not selected, an undefined or revoked user ID returns only the general failure message: return code 8, reason code 2.

Example

In the following example, the PUT method is used to change a user password. On completion, the user password is changed.

```
PUT https://your.company.com/zosmf/authenticate HTTP/1.1
Host: your.company.com
Referer: https://your.company.com/zosmf
Content-Type: application/x-www-form-urlencoded
{"userID" : "ZOSMFT1", "oldPwd" : "QWER1234", "newPwd" : "ZOSMFT1"}

response:
HTTP/1.1 200 OK
response body:
{"returnCode":0,"reasonCode":0,"message":"Success."}
```

Figure 461. Change a user password

Log out of the z/OSMF server

You can use the DELETE method to log out of the z/OSMF server and delete the user's authentication tokens (JSON Web Tokens and LTPA tokens). Your request cookie must include a valid JSON Web Token or LTPA token (or both).

HTTP method and URI path

```
DELETE /zosmf/services/authenticate
```

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

Query parameters

None.

Standard headers

Use the following standard HTTP header with this request:

```
Content-Type: application/x-www-form-urlencoded
```

Customer headers

X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or an empty string (""). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

Request body

None.

Response Body

None.

Required authorizations

See [“Required authorizations” on page 818](#).

Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 4](#).

Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 204 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred.

Table 536. z/OSMF workflow services: operations summary (continued)	
Operation name	HTTP method and URI path
“Get the properties of an archived workflow” on page 1082	GET /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
“Delete an archived workflow” on page 1096	DELETE /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>

Table 537 on page 1021 describes the variables that can be specified in the z/OSMF workflow services URI paths.

Table 537. z/OSMF workflow services: URI path variables	
URI path variable	Description
<version>	The version of the z/OSMF workflow services API. The following value is valid: 1.0.
<workflowKey>	The identifier of a unique instance of a workflow, as returned in the response of the operation that created the workflow.

Using the Swagger interface

You can use the Swagger interface to display information about the z/OSMF workflow REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

Authorization requirements

Use of the z/OSMF workflow services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 3](#).

In addition, the user's z/OS user ID must have:

- READ access to the <SAF-PREFIX>.ZOSMF.WORKFLOW.WORKFLOWS profile in the ZMFAPLA class.
- READ access to the <SAF_PREFIX>.*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF_PREFIX>.IzuManagementFacilityWorkflow.izuUsers resource name in the EJBROLE class.

Error response content

For most 4nn and 5nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 538. Error response body elements for the z/OSMF workflow services API		
Field name	Type	Description
messageID	String	The message identifier identifying the reason for the error.
messageText	String	The message text that describes the error.

Error logging

Errors from the z/OSMF workflow services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

HTTP status codes

The following HTTP status codes are valid:

HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

HTTP 201 Created

The request succeeded and resulted in the creation of an object.

HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

HTTP 204 No content

The request succeeded, but no content is available to be returned.

HTTP 400 Bad request

The request contained incorrect parameters.

HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

HTTP 403 Forbidden

The server received the request, but rejected it.

HTTP 404 Not found

The requested resource does not exist.

HTTP 405 Method not allowed

The requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

HTTP 408 Request timed out

The client did not produce a request within the allowed time. The request can be submitted again later.

HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

HTTP 500 Server error

The server encountered an error when it processed the request. For a more specific indication of the error, check the response for a reason code.

HTTP 501 Not implemented

The request specifies an HTTP method that is not recognized by the server.

HTTP 503 Service unavailable

The request cannot be carried out by the server because of a temporary condition. A suggested wait time might be indicated in a Retry-After header, if one is provided in the response. Otherwise, the requestor can treat the response as a 500 response.

HTTP 504 Gateway timeout

The server, which is acting as a gateway or proxy, did not receive a timely response from the server that was specified in the URI path (for example, HTTP, FTP, LDAP) or an auxiliary server (such as DNS). This access is needed to complete the request. For example, the server was not able to start a remote REXX or UNIX shell interface.

Create a workflow

You can use this operation to create a z/OSMF workflow on a z/OS system.

HTTP method and URI path

```
POST /zosmf/workflow/rest/<version>/workflows
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF workflow service. The following value is valid: `1.0`.

Query parameters

None.

Description

This operation creates a workflow, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content” on page 1023](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new workflow. The URI path for the workflow is provided in the Location response header and a response body is provided, as described in [“Response content” on page 1030](#).

Workflow access type

By default, general information about the workflow and its steps can be viewed by all users of the Workflows task. If you want to restrict access to a workflow or portions of a workflow, you can do so by specifying an *access type* in the request body for your Create Workflow request. The access type determines which users can view the workflow steps and edit the step notes. The access type is specified on the `accessType` property.

The valid values for the `accessType` property are summarized, as follows:

Public

Information about the workflow, including the steps and notes, can be viewed by all users.

Restricted

Information about steps, variables, and notes, is restricted to a subset of users — the workflow owner, step owners, and step assignees. Other users cannot access this information.

Private

Information is restricted to a subset of users, and is further limited among these users. The workflow owner can access information about steps, variables, and notes. Step owners and assignees can retrieve information about the steps for which they are assigned or own, and the associated variables for those steps. Other users cannot access this information.

The `accessType` property is optional. If you omit it from the request body, the workflow is created with public access.

Except for workflow notes and step notes, this information is also available to REST API requesters through the Get Workflow Properties service. For the types of information that are restricted by access type, see [“Get the properties of a workflow” on page 1031](#).

Request content

The request content is expected to contain a JSON object that describes the workflow to be created. [Table 539 on page 1024](#) lists the fields in the JSON object.

Table 539. Request content for the create workflow request

Field name	Type	Required or optional	Description
workflowName	String	Required	Descriptive name for the workflow (up to 100 characters). The name cannot contain the symbols for less-than (<), greater-than (>), or ampersand (&). z/OSMF validates this name to ensure that it is unique across all of the existing workflows.
workflowDefinitionFile	String	Required	<p>Location of the workflow definition file. This file is the primary XML file for the workflow definition.</p> <p>Specify this value, as follows:</p> <ul style="list-style-type: none"> • If the workflow definition file resides in a data set member, specify the fully qualified data set name, including the member name. Ensure that this data set is cataloged. • If the workflow definition file resides in a z/OS UNIX file, specify the fully qualified path name of the file, beginning with the forward slash (/) and including the file name. For example: /usr/lpp/zosmf/samples/workflow_sample_automation.xml.
workflowDefinitionFilesystem	String	Optional	<p>Nickname of the system on which the specified workflow definition file and any related files reside. The Workflows task obtains the workflow files from this system.</p> <p>Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ _ # @). Nicknames are case-sensitive; for example, SYSTEM1 and System1 are unique values.</p> <p>The system can be running in the local sysplex or in another sysplex in your enterprise. If you select a system in a remote sysplex, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system.</p> <p>If you omit this property, the Workflows task checks the z/OSMF system of the local sysplex for the workflow definition file and related files, by default.</p>

Table 539. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
variableInputFile	String	Optional	<p>Specifies an optional properties file that you can use to pre-specify values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify this property, as follows:</p> <ul style="list-style-type: none"> • If the workflow variable input file resides in a data set member, specify the fully qualified data set name, including the member name. Ensure that this data set is cataloged. • If the workflow variable input file resides in a z/OS UNIX file, specify the fully qualified path name of the file, beginning with the forward slash (/) and including the file name. <p>For the format of the contents of the variable input file, see “Providing a workflow variable input file” on page 1170.</p>
variables	Array of objects	Optional	<p>A list of one or more variables for this workflow. The variables that you specify here take precedence over the variables that are specified in the workflow variable input file.</p> <p>Specify this property as a list of name-value objects, for example:</p> <pre>"variables": [{"name": "user_name", "value": "IBMUSER"}, {"name": "file_name", "value": "textfile.txt"}]</pre> <p>If you plan to include an <i>array</i> type variable on the variables property, specify the variable as a list of strings, with each string separated by the following escape character: \. In the following example, the variable array3 is an array variable:</p> <pre>"variables": [{"name": "user_name", "value": "IBMUSER"}, {"name": "file_name", "value": "textfile.txt"}, {"name": "array3", "value": "clust ["Z0SV23T", "DB211T", {"property1": "tt1", "dsname": "TEST.DSNAME.TT1"}]"}]</pre> <p>For more information about array variables, see “Array variables” on page 1168.</p>

Table 539. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
resolveGlobalConflictBy Using	String	Optional	<p>On creation of the workflow, z/OSMF determines whether any of the variables that are supplied in this request (through the variable input file or variables array) would conflict with existing global variables in the Workflows task. In such cases, this property specifies which version of the variable is used, as follows:</p> <ul style="list-style-type: none"> • When set to <code>input</code>, the global variable conflicts are overridden by the variables in specified input file. The global variable value is updated with the input variable value. Use caution with this setting; your selection affects any other workflows that refer to the same global variable. • When set to <code>global</code>, or omitted, the variable value that is supplied with the request (through the variable input file or variables array) is ignored and the current global value is used. <p>The default is <code>global</code>.</p>

Table 539. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
system	String	Required	<p>Nickname of the system on which the workflow is to be created. Use the nickname that is specified for the system definition in the z/OSMF Systems task.</p> <p>The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ _ # @). Nicknames are case-sensitive; for example, SYSTEM1 and System1 are treated as different values.</p> <p>The workflow steps are performed on this system. Any jobs or scripts in the workflow are run on this system. Similarly, any work that you perform manually for the workflow is done on this system.</p> <p>If the workflow is to be created on a system in a remote sysplex:</p> <ul style="list-style-type: none"> • If the system is running z/OSMF, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system. • If the system is not running z/OSMF, it must be associated with the z/OSMF system for that sysplex. If so, set the z/OSMF system as the host system for the system on which the workflow is to be performed. Similarly, you must ensure that the z/OSMF system in the remote sysplex is enabled for single sign-on. Or, you must include a valid user ID and password (in the request body) for basic authentication with the remote system. <p>For more information about defining z/OSMF systems and enabling them for single sign-on, see Defining your systems to z/OSMF (www.ibm.com/docs/en/zos/2.5.0?topic=systems-defining-your-zosmf).</p>
owner	String	Required	<p>User ID of the workflow owner. This user can perform the workflow steps or delegate the steps to other users.</p> <p>Specify a valid user ID, as it is defined to your installation's z/OS security management product, such as RACF. A valid user ID consists of one to eight alphanumeric characters (A-Z, a-z, 0-9, #, \$, and @).</p>
workflowArchiveSAF ID	String	Optional	<p>Indicates who can access the archived workflow, which is archived from the current workflow to a user specified directory.</p> <p>The default value is the current user ID of the workflow owner. This value can only be specified as the workflow owner user ID or a SAF group ID the workflow owner belongs to. All users under the SAF group ID have access to this archived workflow.</p>

Table 539. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
comments	String	Optional	Specifies any information that you want to associate with the creation of this workflow (up to 500 characters). This information is recorded in the workflow history. Consider including a meaningful comment on the workflow, for example: This workflow was created through the z/OSMF workflow services REST interface.
assignToOwner	Boolean	Optional	Indicates whether the workflow steps are assigned to the workflow owner when the workflow is created. If you set this property to true, or omit the property, z/OSMF assigns the steps to the user ID that is specified on the property owner. If you set this property to false, the workflow steps are left unassigned when the workflow is created. The default is true.
accessType	String	Optional	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in “Workflow access type” on page 1023. The following values are valid: <ul style="list-style-type: none"> • Public • Restricted • Private If you omit this property, the workflow is public, by default.
accountInfo	String	Optional	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
jobStatement	String	Optional	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 - 8 characters.
deleteCompletedJobs	Boolean	Optional	For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully. To retain the job, set this property to false, which is the default. If so, the job remains on the JES spool until it is removed by a user or automated process. To conserve space in the JES spool, consider setting this property to true. If you omit this property, the completed job is retained on the JES spool.

Table 539. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
jobsOutputDirectory	String	Optional	For a workflow that submits a job, this property specifies the name of a UNIX directory that is to be used for automatically saving job spool files from the workflow. Specify a valid UNIX file path and directory on the user's system, beginning with a single forward slash ('/'). For example: /u/IBMUSER/jobFiles. If you omit this property, the job spool files are not saved.
autoDeleteOnCompletion	Boolean	Optional	Indicates whether the workflow is automatically deleted from the local system when all of its steps are marked complete or skipped. If so, include this property in your request and set it to <code>true</code> . When a workflow is deleted, it no longer appears in the Workflows table in the Workflows task user interface. You might use this option to avoid reaching the system limit of 200 workflows. Otherwise, if you want to retain the workflow after it is complete, set this property to <code>false</code> or omit the property from your request. Here, the workflow is retained in the Workflows table until you explicitly delete the workflow. If you omit this property, the workflow instance is retained. Note: A called workflow cannot be deleted until all of its calling workflows are either deleted or archived.
targetSystemuid	String	Optional	The user ID to be used for remote system basic authentication.
targetSystempwd	String	Optional	The password to be used for remote system basic authentication.

Authorization requirements

See [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 1030](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body with the reason code that is indicated and associated error message.

Table 540. HTTP error response codes for a create workflow request	
HTTP error status code	Description
HTTP 400 Bad request	<p>The request contained incorrect parameters. For example:</p> <ul style="list-style-type: none"> • The specified workflow name contains errors or is not unique. • Workflow definition file contains errors or does not exist. • Variable input file does not exist. • Validation error. The specified value does not match the validation criteria for one or more of the following properties: variables, system, owner, archiveSAFID or comments. • An incorrect value is specified for the property resolveGlobalConflictByUsing. • The specified workflow contains runAsUser steps that do not have fully valid signatures.
HTTP 401 UNAUTHORIZED	The requester user ID is not permitted to perform the attempted action.
HTTP 403 Forbidden	The requester user ID is not permitted to the workflow definition file or the variable input file.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the service returns the following:

- URI path of the created workflow in the Location response header.
- Response body, which contains a JSON object with details about the workflow. [Table 541 on page 1030](#) lists the fields in the JSON object.

Table 541. Response from a create workflow request	
Field name	Description
workflowKey	Workflow key. A string value, which is generated by z/OSMF to uniquely identify the workflow instance.
workflowDescription	Workflow description. This value is obtained from the element workflowDescription in the workflow definition file.
workflowID	Workflow ID. A short, arbitrary value that identifies the workflow. This value is obtained from the element workflowID in the workflow definition file.
workflowVersion	Version of the workflow definition file. This value is obtained from the element workflowVersion in the workflow definition file.
vendor	Name of the vendor that provided the workflow definition file. This value is obtained from the element vendor in the workflow definition file.

Example HTTP interaction

In [Figure 463 on page 1031](#), a request is submitted to create the workflow AutomationExample on the system SY1.

```

POST /zosmf/workflow/rest/1.0/workflows HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 203
Authorization: Basic em9zbWZhZDp6b3NtZmFk

{
  "workflowName": "AutomationExample",
  "workflowDefinitionFile": "/usr/lpp/zosmf/samples/workflow_sample_automation.xml",
  "system": "SY1",
  "owner": "zosmfad",
  "assignToOwner": true,
  "accessType": "Restricted",
  "deleteCompletedJobs": true
  "autoDeleteOnCompletion": true
}

```

Figure 463. Sample request to create a workflow

A sample response is shown in [Figure 464 on page 1031](#).

```

HTTP/1.1 201 Created
content-length: 210
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
location: /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0
date: Mon, 21 Oct 2019 18:29:55 GMT
content-type: application/json; charset=UTF-8

{
  "vendor": "IBM",
  "workflowDescription": "Sample demonstrating the use of automated steps in workflow.",
  "workflowID": "automationSample",
  "workflowKey": "d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
  "workflowVersion": "1.0"
}

```

Figure 464. Sample response from a create workflow request

Get the properties of a workflow

You can use this operation to retrieve the properties of a z/OSMF workflow.

HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/workflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be queried.

Query parameters

You can specify the following query parameter on this request:

returnData

This optional query parameter is used to request information about the workflow steps and variables. Include one or both of the following attributes on the `returnData` parameter:

steps

Returns an array of step-info objects; one object for each step in the workflow. [Table 545 on page 1040](#) lists the fields in the step-info JSON object.

variables

Returns an array of variable-info objects; one object for each variable that is referenced in the workflow. [Table 550 on page 1050](#) lists the fields in the variable-info JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:

```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

The response data is limited by the access type of the workflow. For more information, see [“Effects of access type on the returned data” on page 1032](#).

Description

This operation retrieves the properties of a z/OSMF workflow. You can optionally expand the returned information through the specification of query parameters. On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 543 on page 1034](#).

For the format of this information, see the JSON objects that are described in [Table 545 on page 1040](#) and [Table 550 on page 1050](#).

Authorization requirements

See [“Authorization requirements” on page 1021](#).

Effects of access type on the returned data

If you include the optional query parameter `returnData` on the request, the operation can return information about the workflow steps or variables, or both. The amount of data that can be retrieved about the workflow steps or variables can be restricted by the workflow *access type*. This value is specified by the workflow owner at workflow creation time.

Generally, a workflow with a public access type is less restricted in the amount of data that is available to the requestor. A workflow with a restricted or private access type is more secure, and requires the caller user ID to be a workflow owner, step owner, or step assignee to use or access areas in the workflow.

All requestors can retrieve the workflow common properties. This data includes the information that is shown in [Table 543 on page 1034](#).

For variables data, the workflow owner can retrieve all of the variable properties for the workflow. The step owner and step assignees can retrieve the variable properties that are associated with steps they own. Other users cannot retrieve this data; requests for details about the variables from these users result in empty arrays being returned.

For steps data, the returned data depends on the type of step data that is requested. The access type allows the workflow creator to distinguish between the following types of steps data.

Step data type	Steps properties	Public access workflow	Restricted access workflow	Private access workflow
Step common properties	<ul style="list-style-type: none"> • title • name • owner • stepNumber • assignees • state • skills • weight • autoEnable • hasCalledWorkflow • userDefined • optional 	This data can be retrieved by all requestors.	This data can be retrieved by: <ul style="list-style-type: none"> • Workflow owner • Step owner and assignees 	This data can be retrieved by: <ul style="list-style-type: none"> • Workflow owner • Step owner or assignee
Step restricted properties	<ul style="list-style-type: none"> • description • prereqStep 	This data can be retrieved by all requestors.	This data can be retrieved by: <ul style="list-style-type: none"> • Workflow owner • Step owner and assignees 	The workflow owner can retrieve this data for all steps. The step owner and assignees can retrieve this data for their steps only.
Step detail properties	<ul style="list-style-type: none"> • calledInstanceURI • calledWorkflowID • calledWorkflowVersion • calledWorkflowMD5 • calledWorkflowDescription • calledWorkflowDefinitionFile • failedPattern • instructions • instructionsSub • isConditionStep • jobInfo • maxLrecl • output • outputSub • outputVariablesPrefix • procName • regionSize • returnCode • saveAsDataset • saveAsDatasetSub • saveAsUnixFile • saveAsUnixFileSub • scriptParameters • submitAs • successPattern • template • templateSub • timeout • variable-references 	The workflow owner can retrieve this data for all steps. The step owner and assignees can retrieve this data for their steps only.	The workflow owner can retrieve this data for all steps. The step owner and assignees can retrieve this data for their steps only.	The workflow owner can retrieve this data for all steps. The step owner and assignees can retrieve this data for their steps only.

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 543 on page 1034](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 542. HTTP error response codes for a get workflow properties request	
HTTP error status code	Description
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.

Other standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 543 on page 1034](#) lists the fields in the JSON object.

Table 543. JSON object that is returned to a get workflow properties request		
Field name	Type	Description
workflowName	String	Descriptive name for the workflow.
workflowKey	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
workflowDescription	String	Description of the workflow.
workflowID	String	Workflow ID. A short, arbitrary value that identifies the workflow.
workflowVersion	String	Version of the workflow definition file.
workflowDefinitionFileMD5Value	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
vendor	String	Name of the vendor that provided the workflow definition file.
owner	String	User ID of the workflow owner.
workflowArchiveSAFID	String	SAF ID who will own the archived workflow after a workflow is archived to a user specified directory.
system	String	Full name of the z/OS system on which the workflow is to be performed. This value is in the format <i>sysplex.sysname</i> .
jobsOutputDirectory	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
category	String	Category of the workflow, which is general, configuration, or provisioning.
productID	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
productName	String	Name of the product or component that is being configured through the workflow.
productVersion	String	Version and release of the product or component that is configured through the workflow.

Table 543. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
percentComplete	Integer	Percentage of the workflow that is completed. z/OSMF calculates this value based on the number of steps in the workflow and the relative weighting value of each step.
isCallable	Boolean	Indicates whether a workflow is eligible to be called by another workflow. For more information, see “Callable workflows” on page 1111.
containsParallelSteps	Boolean	For a parallel-steps workflow, this property is <code>true</code> . If so, the automation ready steps can be run in parallel (concurrently), thus possibly completing more quickly. Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
scope	String	Restricts a workflow to one instance only. The scope attribute can be set to <code>system</code> , <code>sysplex</code> , or <code>none</code> . For more information, see “Setting the workflow scope” on page 1111.
statusName	String	Indicates the current workflow status, as follows: in-progress One or more steps in the workflow are started. complete Workflow is complete. All steps are marked complete or skipped. automation-in-progress Workflow contains an automated step that is running. canceled Workflow is canceled and cannot be resumed. However, you can view its properties or delete it.
deleteCompletedJobs	Boolean	For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully, as follows: <ul style="list-style-type: none">• <code>false</code> means that the job is retained on the JES spool until it is removed by a user or automated process.• <code>true</code> means that the job is deleted from the JES spool after it completes or fails.

Table 543. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
automationStatus	Object	<p>An automation-info object that contains details about the most recent start automation request for the workflow. The content of this property depends on the following factors:</p> <ul style="list-style-type: none"> • If no automation was performed for the workflow, this property is null. • If automation processing is still in progress, this property indicates the step that is being processed. • If automation was restarted after it was stopped, this property indicates the status of the current start automation request. • If automation is stopped and the workflow status is complete, this property indicates that automation is completed. • If automation is stopped and the workflow status is not complete, this property identifies the step that is most closely related to the reason why automation was stopped. <p>Notes about parallel-step workflows:</p> <ul style="list-style-type: none"> – When a parallel-steps workflow is started, all of its automation ready steps are processed until they complete or fail, or automation is stopped. Failure of a step does not stop automation processing for other automation ready steps in the workflow. – In a parallel-steps workflow: <ul style="list-style-type: none"> - The automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. - If automation is currently stopped and the workflow is not yet complete, this property identifies the first uncompleted step that was returned to the Get Properties request. <p>Table 544 on page 1038 lists the fields in the automation-info object.</p>
autoDeleteOnCompletion	Boolean	<p>Specifies whether the workflow is automatically deleted from the system after it completes successfully, as follows:</p> <ul style="list-style-type: none"> • false means that the workflow is retained after it is complete, until it is removed by a user. A complete workflow is one in which all of its steps are marked complete or skipped. • true means that the workflow is automatically deleted from the system after it completes. As a result, the workflow is removed from the Workflows table in the Workflows task user interface.

Table 543. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
access	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in “Workflow access type” on page 1023 . The following values are valid: <ul style="list-style-type: none"> • Public • Restricted • Private
accountInfo	String	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
jobStatement	String	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 - 8 characters.
templateID	String	Specifies the unique identifier for the template. Derived from a workflow internal variable, <code>\${_workflow-templateID}</code> . This property is returned only for a provisioning workflow.
actionID	String	For an actions workflow, this property specifies the action ID for the action object. Derived from a workflow internal variable, <code>\${_workflow-actionID}</code> . For other types of provisioning workflows, this property is null. This property is returned only for a provisioning workflow.
registryID	String	Specifies the ID of the software services registry. Derived from a workflow internal variable, <code>\${_workflow-registryID}</code> . This property is returned only for a provisioning workflow.
parentRegistryID	String	Specifies the ID of the software instance parent registry entry. Derived from a workflow internal variable, <code>\${_workflow-parentRegistryID}</code> . This property is returned only for a provisioning workflow.
domainID	String	Specifies the ID of the domain that is associated with the template. Derived from a workflow internal variable, <code>\${_workflow-domainID}</code> . This property is returned only for a provisioning workflow.
tenantID	String	Specifies the ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-tenantID}</code> . This property is returned only for a provisioning workflow.

Table 543. JSON object that is returned to a get workflow properties request (continued)		
Field name	Type	Description
software-service-instance-name	String	Specifies the created software service instance name. Derived from workflow internal variable, <code>\${_workflow-softwareServiceInstanceName}</code> . This property is returned only for a provisioning workflow.
templateName	String	Specifies the name of the template that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-templateName}</code> . This property is returned only for a provisioning workflow.
globalVariableGroup	String	Global variable group for the workflow.
isInstanceVariableWithoutPrefix	String	Indicates whether the simplified format is used for references to instance variables. If true, variable references are simplified; they omit the prefix <code>instance-</code> . If false, variable references must include the prefix <code>instance-</code> .
steps	Array of objects	Array of one or more step-info objects that contain details about each of the steps in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> . The content of this array depends on what the requestor is permitted to see. For more information, see “Description” on page 1032 . Table 545 on page 1040 lists the fields in the step-info object.
variables	Array of objects	Array of one or more variable-info objects that contain details about the variables that are used in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> . The content of this array depends on what the requestor is permitted to see. For more information, see “Description” on page 1032 . Table 550 on page 1050 lists the fields in the variable-info object.

Format of the automation-info object

Table 544 on page 1038 lists the fields in the automation-info JSON object.

Table 544. Get Workflow Properties request: Format of the automation-info object		
Field name	Type	Description
startUser	String	User ID of the user who initiated the automation processing.
startedTime	Timestamp	Time that automation processing started. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.

Table 544. Get Workflow Properties request: Format of the automation-info object (continued)		
Field name	Type	Description
stoppedTime	Timestamp	Time that automation processing stopped. If automation is still in progress, this property is set to null. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
currentStepName	String	Depending on the current phase of automation processing, this property contains one of the following values: <ul style="list-style-type: none"> Name of the step that is being processed automatically. Name of the step that caused automation to stop. For a workflow that uses parallel processing (a <i>parallel-steps workflow</i>) this value is the name of the first step that is incomplete. If automation is stopped and the workflow status is complete, this property is set to null.
currentStepNumber	String	The step number. If automation is stopped and the workflow status is complete, this property is set to null.
currentStepTitle	String	Step title. If automation is stopped and the workflow status is complete, this property is set to null.
messageID	String	Message identifier for the accompanying message. If automation is still in progress, this property is set to null.
messageText	String	Message text that describes the reason that automation is stopped. If automation is still in progress, this property is set to null.

Format of the step-info object

Table 545 on page 1040 lists the fields in the step-info JSON object. Not all of the properties are returned for every step. Some properties are returned or omitted depending on the step type, as noted in the **When returned** column. This information in this column indicates whether valid data is returned for the step, as follows:

All step types

Properties that are returned for all step types.

Calling steps

Properties that are returned for a step that calls another workflow for execution.

Template steps

Properties that are returned for a step that runs a program, such as a batch job, REXX exec, or UNIX shell script.

REST steps

Properties that are returned for a step that issues a REST API request, such as a GET or PUT request.

Regarding returned data, a template step and a REST step are mutually exclusive. The returned information for a template step does not include the properties for a REST step. Likewise, the returned information for a REST step does not include the properties for a template step. To help you identify which steps are REST steps, the step-info object includes the `isRestStep` property, set to `true` or `false`.

Table 545. Get Workflow Properties request: Format of the step-info object

Field name	Type	When returned	Description
name	String	All step types	Name of the step.
actualStatusCode	String	REST steps only	The actual HTTP status code that is received from the REST API request. To obtain this value, map it to a workflow variable.
assignees	String	Calling steps and template steps	Step assignees. One or more user IDs that are assigned to the step. Multiple items are separated by commas.
autoEnable	Boolean	All step types	Indicates whether the step can be performed automatically when all prerequisite steps are completed, and no user inputs are required.
calledInstanceKey	String	Calling steps only	For a step that calls another workflow for execution, this property contains the key of the called workflow instance. You can use this value to retrieve information about the called workflow. This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
calledInstanceScope	String	Calling steps only	For a step that calls another workflow for execution, this property contains the scope of the called workflow instance. See “Setting the workflow scope” on page 1111 for more information.
calledInstanceURI	String	Calling steps only	For a step that calls another workflow for execution, this property contains the URI path of the called workflow instance. You can use this value to retrieve information about the called workflow. This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
calledWorkflowID	String	Calling steps only	Specifies the workflow ID of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property calledWorkflowMD5 is specified.
calledWorkflowVersion	String	Calling steps only	Specifies the workflow version of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property: <ul style="list-style-type: none"> • Is null when the property calledWorkflowMD5 is specified • Might be null when the property calledWorkflowID is specified.

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
calledWorkflowMD5	String	Calling steps only	Specifies the 128-bit hash value of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property calledWorkflowID is specified.
calledWorkflowDescription	String	Calling steps only	Describes the workflow to be called, from the point of view of the calling workflow.
calledWorkflowDefinitionFile	String	Calling steps only	Specifies the name of the workflow definition file that is used to create a new workflow if an existing instance is not found when this step is performed. This property might be null.
description	String	All step types	Step description.
expectedStatusCode	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
failedPattern	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
hasCalledWorkflow	Boolean	Calling steps and template steps	Indicates whether this step calls another workflow (true or false). If true, this step is a "calling" step, that is, it calls another workflow for execution. If false, it is a template step. This property is returned only when steps=null, which indicates a leaf step.
hostname	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
httpMethod	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> • GET • PUT • POST • DELETE
instructions	String	Template steps only	Detailed instructions on what the user must do to perform the step.
instructionsSub	Boolean	Template steps only	Indicates whether the step instructions contain variables (true or false).

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
isConditionStep	Boolean	Calling steps and template steps	Indicates whether this step is a conditional step (true or false).
isRestStep	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> • actualStatusCode • expectedStatusCode • hostname • hostnameSub • httpMethod • port • portSub • queryParameters • queryParametersSub • requestBody • requestBodySub • schemeName • schemeNameSub • uriPath • uriPathSub <p>The following step properties are not applicable for a REST step and thus, are omitted from the output:</p> <ul style="list-style-type: none"> • template • templateSub • output • outputSub • saveAsDataset • saveAsDatasetSub • saveAsUnixFile • saveAsUnixFileSub • submitAs • maxLrecl • returnCode

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
jobInfo	JSON object	Template steps only	<p>For a step that submits a job, this property contains the jobInfo object, which contains details about the job. Otherwise, this property is null.</p> <p>This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code>.</p> <p>“Format of the jobInfo object” on page 1047 describes the fields in the jobInfo object.</p>
maxLrecl	Integer	Template steps only	For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.
optional	Boolean	All step types	Indicates whether the step is optional (true or false).
output	String	Template steps only	Indicates the name of the output file that is produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
outputSub	Boolean	Template steps only	Indicates whether the output file name contains variable substitution (true or false).
outputVariablesPrefix	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
owner	String	Calling steps and template steps	User ID of the step owner.
port	String	REST steps only	Port number that is associated with the REST request.
portSub	Boolean	REST steps only	Indicates whether the port number contains variable substitution (true or false).
prereqStep	Array of strings	All step types	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
procName	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
queryParameters	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
queryParametersSub	Boolean	REST steps only	Indicates whether the query parameters contain variable substitution (true or false). Otherwise, this property is null.
regionSize	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
requestBody	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.
requestBodySub	Boolean	REST steps only	Indicates whether the request body variable substitution (true or false). Otherwise, this property is null.
returnCode	String	Template steps only	For a step that submits a job to run, this property indicates the return code that was returned when the job was submitted.
runAsUser	String	All step types	<p>The user ID under which the step is to be performed (the runAsUser ID).</p> <p>When the property <code>runAsUserDynamic</code> is true, the runAsUser ID is determined by using variable substitution when the step is performed.</p> <p>This property is omitted if the runAsUser element is not specified for the step.</p>
runAsUserDynamic	Boolean	All step types	<p>Indicates whether the runAsUser ID value can change:</p> <p>true The runAsUser ID value is not final and can change during the processing of the workflow. Its value is determined by using variable substitution when the step is performed.</p> <p>false The runAsUser ID is final and cannot change during the processing of the workflow. Its value is determined when the workflow is created.</p> <p>This property is omitted if the runAsUser element is not specified for the step.</p>
saveAsDataset	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
saveAsDatasetSub	Boolean	Template steps only	Indicates whether the data set name contains variable substitution (true or false).
saveAsUnixFile	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
saveAsUnixFileSub	Boolean	Template steps only	Indicates whether the UNIX file name contains variable substitution (true or false).
schemeName	String	REST steps only	The scheme name that is used for the REST request. For example: http.
schemeNameSub	Boolean	REST steps only	Indicates whether the scheme name contains variable substitution (true or false).
scriptParameters	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
skills	String	Calling steps and template steps	The type of skills that are required to perform the step.

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
state	String	All step types	<p>State of the step. One of the following status indicators is displayed:</p> <ul style="list-style-type: none"> • Unassigned. The step is in the <i>Unassigned</i> state; no users or groups are assigned to the step. • Assigned. Users or groups are assigned to the step, but no user accepted ownership of the step. • Not Ready. User accepted ownership of the step. However, a prerequisite step must be completed or a conditional dependency must be satisfied before the step can be performed. • Ready. The step is ready to be performed; all prerequisite steps and conditional dependencies are satisfied. • In Progress. The step is in progress. For a parent step, a state of <i>In Progress</i> means that at least one of the child steps is started, but is not yet complete, overridden, or skipped. For a leaf step, a state of <i>In Progress</i> means that the step is started, but is not yet complete, overridden, or skipped. • Submitted. The step included a job, which the step owner submitted. • Complete. The step was completed. • Skipped. The step was bypassed by the step assignee. • Complete (Override). The step was marked complete, but the work was performed outside of the Workflows task. • Failed. The step included a job that was submitted by the step owner. However, the job failed to complete successfully. • Conflicts. The step created an output file for use by a subsequent step. However, values in that file conflict with existing instance or global variables. • Condition Not Satisfied. The step is a conditional step, and the condition is not satisfied.
stepNumber	String	All step types	The step number. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.
steps	Array of objects	All step types	<p>For a parent step, this is a nested array of step-info objects. For a leaf step, this property is null.</p> <p>Check this property first before you check the other, non-common step properties. A non-null value here means that the calling step properties are omitted, as are the template step properties and the REST step properties.</p>

Table 545. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
submitAs	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The possible values are the following: <ul style="list-style-type: none"> • "JCL " • "TSO-REXX" • "shell-JCL " • "TSO-REXX-JCL " • "TSO-UNIX-REXX" • "TSO-UNIX-shell "
successPattern	String	Template steps only	Regular expression that is returned for a successful program execution.
template	String	Template steps only	Indicates the template that is used to run a program or batch job (inline or external file).
templateSub	Boolean	Template steps only	Indicates whether template contains variable substitution (true or false). The default is false.
timeout	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.
title	String	All step types	Step title.
uriPath	String	REST steps only	The URI path to use for the REST request.
uriPathSub	Boolean	REST steps only	Indicates whether the URI path contains variable substitution (true or false).
userDefined	Boolean	All step types	Indicates whether the step was added manually to the workflow (true or false). If true, the step was added by the workflow owner, using the Update Workflow Steps action in the Workflows table. If false, the step was defined in the workflow definition that was used to create the workflow.
variable-references	Array of objects	Template steps only	An array of variable-reference objects, the format of which is described in Table 549 on page 1050 .
weight	Integer	Calling steps and template steps	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

Format of the jobInfo object

Table 546 on page 1048 lists the fields in the jobInfo JSON object.

Table 546. Get Workflow Properties request: Format of the jobInfo object		
Field name	Type	Description
jobstatus	JSON object	Contains the jobstatus object, which contains details about the job. Otherwise, this property is null. “Format of the jobstatus object” on page 1048 describes the fields in the jobstatus object.
jobfiles	Array	Contains an array of one or more objects that contain details about each of the files that are created by the job. Otherwise, this property is null. “Format of the jobfiles object” on page 1049 lists the fields in the jobfiles object.

Format of the jobstatus object

Table 547 on page 1048 lists the fields in the jobstatus JSON object.

Table 547. Get Workflow Properties request: Format of the jobstatus object		
Field name	Type	Description
retcode	String	Job completion code. One of the following values: ABENDUnnnn Job ended with the user abend code <i>nnnn</i> . ABEND Sxxx Job ended with the system abend code <i>xxx</i> . CANCELED Job was canceled. CC nnnn Job ended with the completion code <i>nnnn</i> . CONV ABEND Converter ended abnormally when processing the job. CONV ERROR Converter error when processing the job. JCL ERROR Job encountered a JCL error. SEC ERROR Job failed a security check. SYS FAIL System failure. If this value is null, the job has not yet completed.
jobname	String	Job name.

Table 547. Get Workflow Properties request: Format of the jobstatus object (continued)		
Field name	Type	Description
status	String	Job status. One of the following values: INPUT Job is in input processing. ACTIVE Job is running. OUTPUT Job is on the hardcopy output queue. If this value is null, the job status could not be determined.
owner	String	The z/OS user ID associated with the job.
subsystem	String	The primary or secondary JES subsystem. If this value is null, the job was processed by the primary subsystem.
class	String	Job execution class.
type	String	Job type. One of the following values: JOB Batch job. STC Started task. TSU TSO/E user.
jobid	String	Job identifier.

Format of the jobfiles object

Table 548 on page 1049 lists the fields in the jobfiles JSON object.

Table 548. Get Workflow Properties request: Format of the jobfiles object		
Field name	Type	Description
id	Integer	Data set number (key).
ddname	String	DDNAME for the data set creation.
byte-count	Integer	Number of bytes on spool that is consumed by the spool file. The value can be zero (0).
record-count	Integer	Number of records in the spool file. The value can be zero (0).
class	String	Class that is assigned to the spool file.
stepname	String	Step name for the step that created this data set. The value can be null.
procstep	String	Procedure name for the step that created this data set. The value can be null.

Format of the variable-reference object

Table 549 on page 1050 lists the fields in the variable-reference JSON object.

Table 549. Get Workflow Properties request: Format of the variable-reference object		
Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.

Format of the variable-info object

Table 550 on page 1050 lists the fields in the variable-info JSON object.

Table 550. Get Workflow Properties request: Format of the variable-info object		
Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.
type	String	Type of variable, which is one of the following values: <ul style="list-style-type: none"> • boolean • string • number • date • time • array
value	String	Variable value.
visibility	String	Public or private.

Example HTTP interaction

In the following example, the GET method is used to retrieve information about a workflow. The workflow is uniquely identified by the workflow key, which is represented by the following string value: 26f1fd84-058b-443c-8e06-5ec318ecdb86. The query parameter `returnData=steps,variables` is included to request more information about the workflow steps and variables.

```
GET /zosmf/workflow/rest/1.0/workflows/26f1fd84-058b-443c-8e06-5ec318ecdb86?returnData=steps,variables
HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 465. Sample request to get workflow properties

An example of the response is shown in the figures that follow.


```

HTTP/1.1 200 OK
content-length: 9155
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Thur, 12 Mar 2020 18:30:33 GMT
content-type: application/json; charset=UTF-8
{
  "access": "Public",
  "productID": "ABC123",
  "jobStatement": null,
  "deleteCompletedJobs": false,
  "productName": "Product ABC",
  "globalVariableGroup": null,
  "productVersion": "Version 1",
  "jobsOutputDirectory": null,
  "vendor": "IBM",
  "scope": "none",
  "statusName": "in-progress",
  "workflowID": "programExecutionSample",
  "owner": "zosmfad",
  "accountInfo": null,
  "isInstanceVariableWithoutPrefix": false,
  "variables": [
    {
      "visibility": "private",
      "scope": "instance",
      "name": "procNameVariable",
      "type": "string",
      "value": null
    },
    {
      "visibility": "private",
      "scope": "instance",
      "name": "st_group",
      "type": "string",
      "value": null
    },
    {
      "visibility": "private",
      "scope": "instance",
      "name": "st_user",
      "type": "string",
      "value": null
    }
  ],
  "workflowName": "testProgramExecutionSample",
  "automationStatus": null,
  "autoDeleteOnCompletion": false,
  "percentComplete": 0,
  "workflowDescription": "Sample that demonstrates how to run an executable program from a step.\n\t",
  "steps": [
    {
      "template": "#!/bin/sh\nnecho \"this is a sample to submit a shell script to run immediately\"\nnecho  
\"the first parameter is :\" \"$1\" \t\necho ${instance-st_user}  
\necho prefix:st_group = SYS123\nnecho prefix:st_user = USERS\nnecho  
\"This symbol is used to indicate success\"\n\t\necho  
\"The program ran successfully!\"",
      "instructions": "This step outputs some variables and prints a few words.\n\t",
      "maxLrecl": 1024,
      "failedPattern": [
        "failed.*"
      ],
      "assignees": "zosmfad",
      "description": "In this step, you submit an inline UNIX shell script for immediate processing \n\t\tton the host system.  
  
In this example, the step is expected to complete successfully.\n\t\t\t",
      "outputVariablesPrefix": "prefix:",
      "variable-references": [
        {
          "scope": "instance",
          "name": "st_group"
        },
        {
          "scope": "instance",
          "name": "st_user"
        },
        {
          "scope": "instance",
          "name": "procNameVariable"
        }
      ]
    }
  ]
}

```

Figure 466. Sample response from a get workflow properties request (Part 1 of 3)

List the workflows for a system or sysplex

You can use this operation to list the z/OSMF workflows for a system or sysplex.

HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/workflows
```

In this request, the URI path variable *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.

Query parameters

Optionally, your request can include one or more of the following query parameters to filter the results:

workflowName

Workflow name. You can specify a regular expression here to match desired workflow names.

category

Category of the workflow, which is either general or configuration.

system

Nickname of the system on which the workflow is to be performed.

statusName

Workflow status, which can be one of the following values:

- in-progress
- complete
- automation-in-progress
- canceled

owner

Workflow owner (a valid z/OS user ID).

vendor

Name of the vendor that provided the workflow definition file.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark (?) to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).

Description

This operation retrieves a list of workflows that match your search criteria. You can filter the returned list of workflows through the specification of query parameters. You can, for example, limit the results to workflows by name, or the workflows for a particular z/OS system.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 551 on page 1055](#).

Authorization requirements

See [“Authorization requirements” on page 1021](#).

For the requestor of this service, the response data is limited by the workflow *access type*. This value is selected by the workflow owner at workflow creation time. All workflows with a public access type are

included in the response. To list restricted or private access type workflows, however, the requestor user ID must be a workflow owner, step owner, or step assignee.

For more information, see [“Workflow access type” on page 1023](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, a non-successful HTTP status code is returned and the response body is a standard error response body with the message ID and the associated error message. For the codes that can be returned, see [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body contains one property, called workflows. This property is an array of workflow-info objects. [Table 551 on page 1055](#) lists the fields in the workflow-info object. If no workflows match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 551. List workflows request: Format of the workflow-info object		
Field name	Type	Description
workflowName	String	Descriptive name for the workflow.
workflowKey	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
workflowDescription	String	Description of the workflow.
workflowID	String	Workflow ID. A short, arbitrary value that identifies the workflow.
workflowVersion	String	Version of the workflow definition file.
workflowDefinitionFileMD5Value	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
instanceURI	String	Workflow instance URI path, which you can use to retrieve information about the workflow.
owner	String	User ID of the workflow owner.
vendor	String	Name of the vendor that provided the workflow definition file.
access	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in “Workflow access type” on page 1023 . The following values are valid: <ul style="list-style-type: none">• Public• Restricted• Private

Example HTTP interaction

In the following example, the GET method is used to list the workflows on a system. Here, the query parameter workflowName is included to limit the results to workflows with names that begin with AutomationExample.

```
GET /zosmf/workflow/rest/1.0/workflows?workflowName=AutomationExample.* HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 469. Sample request to list workflows

For a successful request, the HTTP response includes a JSON document that contains the requested information. In the following example, one workflow was found to match the request query parameter.

```
HTTP/1.1 200 OK
content-length: 464
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8

{
  "workflows": [
    {
      "instanceURI": "/zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
      "owner": "zosmfad",
      "vendor": "IBM",
      "workflowDefinitionFileMD5Value": "a8825b7497793bc620b0edffa8b97cd9",
      "workflowDescription": "Sample demonstrating the use of automated steps in workflow.",
      "workflowID": "automationSample",
      "workflowKey": "d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
      "workflowName": "AutomationExample|Canceled|1423679433714",
      "workflowVersion": "1.0",
      "access": "Public"
    }
  ]
}
```

Figure 470. Sample response from a list workflows request

Start a workflow

You can use this operation to start a z/OSMF workflow on a z/OS system. The workflow to be started must contain at least one automated step.

HTTP method and URI path

```
PUT /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/start
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be started.

Query parameters

None.

Description

This operation is used to start an automated workflow, that is, a workflow with at least one step that can be performed automatically. For information about designing an automated workflow, see [“Automated steps”](#) on page 1143.

On successful completion, HTTP status code 202 (Accepted) is returned. If you specify a notification URL in your request, a JSON object is posted to the URL when automation stops; the format of the object is described in [“Content that is posted to the notification URL”](#) on page 1060.

This request must be submitted from the user ID of the step owner for the automated step. Otherwise, the request ends with HTTP status code 400 and an error message. Also, if used with a workflow that contains no automated steps, the request ends with HTTP status code 400 and an error message.

Parallel-steps workflows

A workflow with steps that can be run in parallel (concurrently) is called a *parallel-steps workflow*. When you start a parallel-steps workflow, the Workflows task locates the automation ready steps and attempts to run them concurrently.

A step is considered to be *automation ready* when it is:

- Enabled for automation. In the workflow definition file, the attribute `autoEnable=true` is specified on the step element (`<step>`).
- In an eligible state: *ready*, *in-progress*, or *failed*. For a failed step, the Workflows task performs the step again.

To use parallel processing, an automated workflow must include the attribute `parallelSteps=true` in the workflow metadata. Otherwise, if this attribute is set to `false` or omitted, automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

In a parallel-steps workflow, the failure of an automated step does not stop automation processing for the other automated steps. Processing continues until all of the automated steps are completed or failed, or a condition occurs that stops automation processing, such as a user stopping automation by using the **Stop Automation** action in the Workflows task.

Note: The ability to suspend step processing is mutually exclusive with the ability to run steps in parallel. Therefore, if a workflow includes the suspend element (`<suspend>`) in the step definition, it is not eligible for parallel processing. The Workflows task enforces this restriction. An attempt to start a workflow that contains both the suspend element (`<suspend>`) and the attribute `parallelSteps=true` will result in an error.

Request content

Your request can contain a JSON object that specifies options for starting the workflow. The request content is optional; it can be omitted or be empty. If so, default values are used for any optional properties that can be specified.

[Table 552 on page 1058](#) lists the fields in the JSON object.

Table 552. Request content for the start workflow request

Field name	Type	Required or optional	Description
resolveConflictByUsing	String	Optional	<p>Indicates how variable conflicts, if any, are to be handled when the Workflows task reads in the output file from a step that runs a REXX exec or UNIX shell script.</p> <p>The following values are valid for this parameter:</p> <p>outputFileValue Allow the output file values to override the existing values. This setting applies to instance variables only; global variables are not overridden by variables in the output file.</p> <p>existingValue Use the existing variables values instead of the output file values.</p> <p>leaveConflict Automation is stopped. The user must resolve the conflict manually.</p> <p>If you omit this property, the default is <code>outputFileValue</code>.</p>
stepName	String	Optional	<p>The name of the step at which automation is to begin. If you omit this property, automation processing begins with the first step in the workflow.</p> <p>In a parallel-steps workflow, the automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. Therefore, automation cannot be directed to start at a specific step. If you specify this property for a parallel-steps workflow, the property is accepted but ignored.</p>
performSubsequent	Boolean	Optional	<p>If the workflow contains any subsequent automated steps, this property indicates whether z/OSMF is to perform the steps. If you set this property to <code>true</code>, or omit the property, z/OSMF attempts to perform the automated steps. If you set this property to <code>false</code>, z/OSMF attempts to perform the specified step only. The default is <code>true</code>.</p> <p>In a parallel-steps workflow, the automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. Therefore, automation cannot be directed to start at a specific step. If you specify this property for a parallel-steps workflow, the property is accepted but ignored.</p>

Table 552. Request content for the start workflow request (continued)			
Field name	Type	Required or optional	Description
notificationUrl	String	Optional	<p>A notification URL (up to 2000 characters).</p> <p>Depending on the requirements of your application, you might want to receive a notification when the automated step or steps reach an end state (either completion or failure). Suppose, for example, that your application includes a servlet that is to be given control on completion of the automated steps. If so, you can optionally specify a notification URL to be posted when automation ends.</p> <p>Your installation must add this URL to the list of allowable sites. Work with your security administrator to create the appropriate authorization; see “Allowing cross-site access to REST services” on page 5.</p> <p>If specified, this URL receives a JSON object with the result of the automation processing. For the format of the JSON object, see Table 554 on page 1060.</p>
targetSystemuid	String	Optional	The user ID to be used for remote system basic authentication.
targetSystempwd	String	Optional	The password to be used for remote system basic authentication.

Authorization requirements

See [“Authorization requirements”](#) on page 1021.

HTTP status codes

On successful completion, HTTP status code 202 (Accepted) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 553. HTTP error response codes for a start workflow request	
HTTP error status code	Description
HTTP 400 Bad request	<p>The request contained incorrect parameters. For example:</p> <ul style="list-style-type: none"> Incorrect value is specified for the property <code>resolveConflictByUsing</code>. Specified workflow contains no automated steps. Request is submitted from a user ID that is not the step owner for the automated step. For a parallel-steps workflow, the request is submitted from a user ID that is not the step owner for any of the automated steps in the workflow. Automation processing attempts to begin at a step that is not automated. Incorrect value is specified for the property <code>stepName</code>.

<i>Table 553. HTTP error response codes for a start workflow request (continued)</i>	
HTTP error status code	Description
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	Request cannot be performed automatically because the specified workflow has one of the following statuses: <ul style="list-style-type: none"> Automation-in-progress Canceled Complete (for a parallel-steps workflow). A start request is not applicable for a parallel-steps workflow that is complete.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Content that is posted to the notification URL

If you specified a notification URL in your request, the URL receives a JSON object when automation stops. [Table 554 on page 1060](#) lists the fields in the JSON object.

<i>Table 554. Structure of the JSON object that is returned to the notification URL</i>		
Field name	Type	Description
instanceURI	String	Specifies the URI path of the workflow that was being automated. You can use the URI path to retrieve more details about the workflow.
statusName	String	Indicates the current workflow status after automation has stopped. The following are the possible values that might be returned after a stopped automation: in-progress The workflow automation is stopped. The reason is indicated in the messageID and messageText properties. complete Workflow is complete. All steps are marked complete or skipped.
startUser	String	User ID of the user who initiated the automation processing.
startedTime	Timestamp	Time that automation processing started.
stopUser	String	User ID of the user who stopped automation.
stoppedTime	Timestamp	Time that automation processing stopped.
currentStepName	String	Name of the step that caused automation to stop. If the workflow status is complete, this property is set to null.
currentStepNumber	String	The step number. If the workflow status is complete, this property is set to null.

Table 554. Structure of the JSON object that is returned to the notification URL (continued)		
Field name	Type	Description
currentStepTitle	String	Step title. If the workflow status is complete, this property is set to null.
messageID	String	This property contains the message identifier that helps identify the reason that automation is stopped.
messageText	String	This property contains the message text associated with the message identifier found in messageID.

Example HTTP interaction

In Figure 471 on page 1061, a request is submitted to start a workflow. Here, the workflow is identified by the workflow key, which is the following string value: d043b5f1-adab-48e7-b7c3-d41cd95fa4b0. In this example, the request content is empty. As a result, default values are used for any properties that can be specified; see Table 552 on page 1058.

```
PUT /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0/operations/start HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 3
Authorization: Basic em9zbWZhZDp6b3NtZmFk

{}
```

Figure 471. Sample request to start a workflow

A sample response is shown in Figure 472 on page 1061.

```
HTTP/1.1 202 Accepted
content-length: 0
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:29:55 GMT
content-type: application/json; charset=UTF-8
```

Figure 472. Sample response from a start workflow request

Cancel a workflow

You can use this operation to cancel a z/OSMF workflow on a z/OS system.

HTTP method and URI path

```
PUT /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/cancel
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be canceled.

Query parameters

None.

Description

This operation is used to cancel a workflow. Canceling a workflow does not undo any actions that were already performed on the system as part of the workflow.

When canceled, the workflow cannot be resumed. You can view the workflow properties through a GET request, as described in [“Get the properties of a workflow” on page 1031](#). Also, you can delete a canceled workflow, as described in [“Delete a workflow” on page 1063](#).

When canceled, the name of workflow is changed. The name is appended with the text |Canceled|, followed by a timestamp (the date and time expressed in milliseconds since midnight on January 1, 1970 UTC). An example is shown in [“Example HTTP interaction” on page 1062](#).

This request is failed for a workflow with the status *Automation in Progress*. That is, it is not possible to cancel the workflow while an automated step is running. You must either allow the processing to complete, or you can stop the processing through the Workflows task.

Request content

None.

Authorization requirements

This request is available to the workflow owner only. A cancel request from another user is rejected with HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned. Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 555. HTTP error response codes for a cancel workflow properties request	
HTTP error status code	Description
HTTP 403 Forbidden	The request was submitted from a user ID that is not the workflow owner.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	The request cannot be processed because the workflow has the status <i>Automation in Progress</i> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body contains one property, workflowName, which specifies the new name of the canceled workflow.

Example HTTP interaction

In the following example, the PUT method is used to cancel an instance of a workflow. Here, the workflow is identified by the workflow key, which is the following string value: d043b5f1-adab-48e7-b7c3-d41cd95fa4b0.

```
PUT /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0/operations/cancel HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 0
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 473. Sample request to cancel a workflow

For a successful request, HTTP response code 200 is returned with the canceled workflow name in response body.

```
HTTP/1.1 200 OK
content-length: 59
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:33 GMT
content-type: application/json; charset=UTF-8

{
  "workflowName": "AutomationExample|Canceled|1423679433714"
}
```

Figure 474. Sample response from a cancel workflow request

Delete a workflow

You can use this operation to remove a z/OSMF workflow from a z/OS system.

HTTP method and URI path

```
DELETE /zosmf/workflow/rest/<version>/workflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- *<workflowKey>* identifies the workflow to be deleted.

Query parameters

None.

Description

This operation is used to delete a workflow from z/OSMF, including any notes that accompany the workflow and its steps, and the history log for the workflow. Deleting a workflow does not undo any actions that were performed on the system as part of the workflow. If you delete a workflow, you are responsible for undoing manually any changes on the system that you no longer require. Ensure that all applicable back-out procedures are followed. See your workflow provider for this information.

This request is failed for a workflow with the status *Automation in Progress*. That is, it is not possible to delete the workflow while an automated step is running. You must either allow the processing to complete, or you can stop the processing through the Workflows task.

You cannot delete a *called workflow*, that is, a workflow that is in-progress as a result of being called by another workflow for execution. For design considerations for called workflows, see [“Calling steps” on page 1137](#).

Request content

None.

Authorization requirements

For a general workflow or configuration workflow, the ability to delete the workflow is limited to the current workflow owner and members of the z/OSMF workflow administrators group. For a provisioning workflow, the domain administrator is also able to delete a workflow. A delete request from another user is rejected with the HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 556. HTTP error response codes for a delete workflow request	
HTTP error status code	Description
HTTP 403 Forbidden	The request was submitted from a user ID that is not the workflow owner.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	The request cannot be processed because the workflow has the status <i>Automation in Progress</i> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

None.

Example HTTP interaction

In the following example, the DELETE method is used to delete a workflow. Here, the workflow is identified by the workflow key, which is the following string value: d043b5f1-adab-48e7-b7c3-d41cd95fa4b0.

```
DELETE /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0 HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 475. Sample request to delete a workflow

For a successful request, the HTTP response 204 is returned.

```
HTTP/1.1 204 No Content
content-length: 0
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8
```

Figure 476. Sample response from a delete workflow request

Retrieve a workflow definition

You can use this operation to retrieve the contents of a z/OSMF workflow definition from a z/OS system.

HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/workflowDefinition
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF workflow service. The following value is valid: `1.0`.

Query parameters

You can specify the following query parameters on this request:

definitionFilePath

Specifies the location of the workflow definition file, which is either a UNIX path name (including the file name) or a fully qualified z/OS data set name. This parameter is required.

workflowDefinitionFileSystem

Nickname of the system on which the specified workflow definition file and any related files reside. The Workflows task obtains the workflow files from this system.

Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ _ # @). Nicknames are case-sensitive; for example, `SYSTEM1` and `System1` are unique values.

The system can be running in the local sysplex or in another sysplex in your enterprise. If you select a system in a remote sysplex, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system.

If you omit this parameter, the Workflows task checks the z/OSMF system of the local sysplex for the workflow definition file and related files, by default.

returnData

Use this optional parameter to request more information about the workflow definition file. Include one or both of the following attributes on the `returnData` parameter:

steps

Returns an array of step-definition objects; one object for each step in the workflow. [Table 559 on page 1068](#) lists the fields in the step-definition JSON object.

variables

Returns an array of variable-definition objects, with one object for each variable that is referenced in the workflow. [Table 563 on page 1074](#) lists the fields in the variable-definition JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:

```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

Description

This operation retrieves the content of a z/OSMF workflow definition. You can optionally expand the returned information through the specification of query parameters.

A workflow definition might consist of multiple XML files, including a primary file and possibly other files that are included by the primary file. The workflow definition resides in a z/OS UNIX file system or a z/OS data set.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 558 on page 1066](#). If you include the optional query parameter `returnData` on the request, the operation can return more information about the workflow definition steps or variables, or both. For the format of this information, see the JSON objects that are described in [Table 559 on page 1068](#) and [Table 563 on page 1074](#).

Authorization requirements

See [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 558 on page 1066](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the indicated reason code and associated error message.

Table 557. HTTP error response codes for a retrieve a workflow definition request	
HTTP error status code	Description
HTTP 403 Forbidden	The requester user ID is not permitted to the workflow definition file.
HTTP 404 Not found	The specified workflow definition file was not found. This resource is specified on the query parameter <code>definitionFilePath</code> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 558 on page 1066](#) lists the fields in the object.

Table 558. JSON object that is returned to a retrieve a workflow definition request		
Field name	Type	Description
workflowDefaultName	String	Identifies the default name for the workflow. This value is shown in the workflow name field of the Workflows task when the user creates a new workflow instance.
workflowDescription	String	Description of the workflow.
workflowID	String	Workflow ID. A short, arbitrary value that identifies the workflow.
workflowVersion	String	Version of the workflow definition file.
vendor	String	Name of the vendor that provided the workflow definition file.

Table 558. JSON object that is returned to a retrieve a workflow definition request (continued)

Field name	Type	Description
workflowDefinitionFileMD5Value	String	A 128-bit hash value that z/OSMF generates to uniquely identify the workflow definition file.
isCallable	String	Indicates the callable scope for the workflow, as follows: system An instance of this workflow can be called only from a workflow in the same system. sysplex An instance of this workflow can be called from a workflow anywhere in the sysplex. This property is null when the workflow cannot be called by another workflow.
containsParallelSteps	Boolean	For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, this property is <code>true</code> . Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
scope	String	Indicates the singleton scope for the workflow, as follows: system A maximum of one instance of this workflow can exist on any one system in the sysplex. sysplex A maximum of one instance of this workflow can exist in the sysplex. none An existing instance cannot be used. A new instance of this workflow is always created.
jobsOutputDirectory	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
category	String	Category of the workflow, which is general, configuration, or provisioning.
productID	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
productName	String	Name of the product or component that is being configured through the workflow.
productVersion	String	Version and release of the product or component that is configured through the workflow.
globalVariableGroup	String	Global variable group for the workflow.
isInstanceVariableWithoutPrefix	Boolean	Indicates whether the simplified format is used for references to instance variables. If <code>true</code> , variable references are simplified; they omit the prefix <code>instance-</code> . If <code>false</code> , variable references must include the prefix <code>instance-</code> .

Table 558. JSON object that is returned to a retrieve a workflow definition request (continued)		
Field name	Type	Description
steps	Array of objects	Array of one or more step-definition objects that contain details about each of the steps in the workflow definition file. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> . See the “Format of the step-definition object” on page 1068 section for a list of the fields in the step-definition object.
variables	Array of objects	Array of one or more variable-definition objects that contain details about the variables that are defined or referenced in the workflow definition file. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> . Table 563 on page 1074 lists the fields in the variable-definition object.

Format of the step-definition object

The following tables list the fields in the step-definition object:

- Table 559 on page 1068
- Table 560 on page 1068
- Table 561 on page 1069

Table 559. Retrieve a workflow definition request: Fields included in every step-definition object		
Field name	Type	Description
name	String	Name of the step.
title	String	Step title.
description	String	Step description.
prereqStep	Array of strings	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
optional	Boolean	Indicates whether the step is optional (true or false).
steps	Array of objects	For a parent step, this property is an array of one or more step-definition objects that contain details about each of the steps in the workflow. For a leaf step, this property is null.

Table 560. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow)		
Field name	Type	Description
calledWorkflowDescription	String	For a step that calls another workflow for execution, this property contains the description of the called workflow. This information might include details such as the name and location of the workflow definition file that is used to create the called workflow.

Table 560. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow) (continued)

Field name	Type	Description
calledWorkflowID	String	For a step that calls another workflow for execution, this property contains the workflow ID for the called workflow.
calledWorkflowMD5	String	For a step that calls another workflow for execution, this property contains the 128-bit hash value that can be used to identify the called workflow.
calledWorkflowDefinitionFile	String	For a step that calls another workflow for execution, this property contains the path name of the workflow definition file for the called workflow.
calledWorkflowVersion	String	For a step that calls another workflow for execution, this property contains the version of the workflow definition file for the called workflow.
callingStepAutoEnable	Boolean	For a step that calls another workflow for execution, this property indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required (true or false).
callingStepWeight	Integer	For a step that calls another workflow for execution, this property indicates the relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).
callingStepSkills	String	For a step that calls another workflow for execution, this property indicates the type of skills that are required to perform the step.

Table 561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step

Field name	Type	When returned	Description
actualStatusCode	String	REST steps only	The actual HTTP status code that is received from the REST API request. To obtain this value, map it to a workflow variable.
approvers	Array of objects	All step types	An array of objects that contain: approver One or more user IDs that are separated by spaces and can provide approval. approverSub This can be set to true or false regardless of whether the approver field uses substitution.
autoEnable	Boolean	All step types	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required (true or false).

Table 561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
expectedStatusCode	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
failedPattern	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
hostname	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
httpMethod	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> • GET • PUT • POST • DELETE
instructions	String	All step types	Detailed instructions on what the user must do to perform the step.

Table 561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
isRestStep	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> • actualStatusCode • expectedStatusCode • hostname • httpMethod • port • propertyMappings • queryParameters • requestBody • schemeName • uriPath <p>The following step properties are not applicable for a REST step, and are therefore omitted from the output:</p> <ul style="list-style-type: none"> • template • output • saveAsDataset • saveAsUnixFile • submitAs • maxLrecl
maxLrecl	Integer	Template steps only	For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.
output	String	Template steps only	Indicates the default name of the output file that is produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
outputVariablesPrefix	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
port	String	REST steps only	Port number that is associated with the REST request.

Table 561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
procName	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log in to the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.
propertyMappings	Array of objects	REST steps only	<p>An array of property mappings, the format of which is:</p> <pre>{ "mapFrom": "property", "mapTo": "variable" },</pre> <p>In the mappings:</p> <p>mapFrom Is the property from the REST request. The value of this property is assigned to the specified "mapTo" workflow variable.</p> <p>mapTo Is the workflow variable that is assigned the value from "mapFrom" property.</p>
queryParameters	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
regionSize	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
requestBody	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.
saveAsDataset	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.
saveAsUnixFile	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
schemeName	String	REST steps only	The scheme name that is used for the REST request. For example: http.
scriptParameters	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
skills	String	All step types	The type of skills that are required to perform the step.

Table 561. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
submitAs	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The following values are valid: <ul style="list-style-type: none"> • "JCL " • "TSO-REXX" • "shell-JCL " • "TSO-REXX-JCL " • "TSO-UNIX-REXX" • "TSO-UNIX-shell "
successPattern	String	Template steps only	Regular expression that is returned for a successful program execution.
template	String	Template steps only	Indicates the template that is used for a JCL job, a REXX exec program, or a UNIX shell script.
timeout	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it times out.
uriPath	String	REST steps only	The URI path to use for the REST request.
variable-specifications	Array of objects	All step types	An array of variable-specification-info objects, the format of which is described in Table 562 on page 1073 .
weight	Integer	All step types	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

Format of the variable-specification-info object

[Table 562 on page 1073](#) lists the fields in the variable-specification-info object.

Table 562. Retrieve a workflow definition request: Format of the variable-specification-info object

Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.
required	Boolean	Indicates whether the variable is required (true or false).

Format of the variable-definition object

[Table 563 on page 1074](#) lists the fields in the variable-definition object.

Table 563. Retrieve a workflow definition request: Format of the variable-definition object

Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.
abstract	String	A brief description of the variable.
category	String	Name of the logical grouping to which the variable belongs.
choice	Array of strings	The choice value for the variable.
decimalPlaces	Integer	Maximum number of decimal places that can be specified.
default	String	Default value of the variable.
description	String	Description of the variable.
exposeToUser	Boolean	Indicates whether the variable is displayed to the user in the Workflows task.
maxLength	Integer	Maximum length of the variable value.
maxValue	String	Maximum value of the variable.
minLength	Integer	Minimum length of the variable value.
minValue	String	Minimum value of the variable.
promptAtCreate	Boolean	Indicates whether the user is prompted to specify a value for the variable during the create workflow process.
regularExpression	String	Provides a standard regular expression that constrains the variable value, as an alternative to the available validation types.
requiredAtCreate	Boolean	Indicates whether a value must be specified for the variable during the create workflow process.
type	String	Type of variable.
validationType	String	Specifies the validation type for the variable.
valueMustBeChoice	Boolean	Indicates whether the variable value must come from the provided choices. If true, the user must choose from the predefined values. If false, the user can enter a custom value.
visibility	String	Indicates whether the variable is displayed to the Workflows task user (either public or private).

Example HTTP interaction

In the following example, the GET method is used to retrieve a workflow definition. The location of the workflow definition is specified on the query parameter `definitionFilePath`. The query parameter `returnData=steps,variables` is included to request more information about the workflow steps and variables.


```

GET
/zosmf/workflow/rest/1.0/workflowDefinition
?definitionFilePath=/usr/lpp/zosmf/samples/workflow_sample_program_execution.xml
&returnData=steps,variables HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk

```

Figure 477. Sample request to get a workflow definition

An example of the response is shown in the figures that follow.

```

HTTP/1.1 200 OK
content-length: 5822
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Thur, 12 Mar 2020 18:30:34 GMT
content-type: application/json; charset=UTF-8
{
  "workflowDefaultName": null,
  "isInstanceVariableWithoutPrefix": false,
  "variables": [
    {
      "visibility": "private",
      "exposeToUser": false,
      "promptAtCreate": false,
      "description": "This value is used to specify a procedure name (proc name) for the TSO/E address space \n\t
        that is used to run the program.\n\t",
      "abstract": "Enter a procedure name for running the program.",
      "type": "string",
      "requiredAtCreate": false,
      "default": null,
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "procNameVariable",
      "category": "TSO procName",
      "choice": null
    },
    {
      "visibility": "private",
      "exposeToUser": false,
      "regularExpression": "^[A-Z$#@]{1}[0-9A-Z$#@]{0,7}$",
      "promptAtCreate": false,
      "validationType": "GROUP",
      "description": "The group name under whose authority the started task will run.\n\t",
      "abstract": "Group name for the started task.",
      "type": "string",
      "requiredAtCreate": false,
      "default": "SYS1",
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "st_group",
      "category": "Started",
      "choice": null
    },
    {
      "visibility": "private",
      "exposeToUser": false,
      "regularExpression": "[0-9A-Z$#@]{1,8}$",
      "promptAtCreate": false,
      "validationType": "USERID",
      "description": "The user ID under whose authority the new started task will run.\n\t",
      "abstract": "User ID for the started task.",
      "type": "string",
      "requiredAtCreate": false,
      "default": "MYSTUSER",
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "st_user",
      "category": "Started",
      "choice": null
    }
  ],
}

```

Figure 478. Sample response from a get workflow definition request (Part 1 of 3)

```

"productID": "ABC123",
"workflowDescription": "Sample that demonstrates how to run an executable program from a step.\n\t",
"steps": [
  {
    "template": "\n#!/bin/sh\nnecho \"this is a sample to submit shell script to run immediately\"\n\nnecho \"the first parameter is :\" $1 \t\n\nnecho ${instance-st_user}\necho prefix:st_group = SYS123\nnecho prefix:st_user = USERS\nnecho \"This symbol is used to indicate success\"\n\nnecho \"The program ran successfully !!\"\n\n",
    "instructions": "This step outputs some variables and prints a few words.\n",
    "autoEnable": "false",
    "maxLrecl": 1024,
    "submitAs": "TSO-UNIX-shell",
    "failedPattern": [
      "failed.*"
    ],
    "description": "In this step, you submit an inline UNIX shell script for immediate processing \n\t\nton the host system. In this example, the step is expected to complete successfully.\n\t\t",
    "weight": 1,
    "outputVariablesPrefix": "prefix:",
    "optional": false,
    "procName": "${instance-procNameVariable}",
    "title": "A step that runs a UNIX shell script.",
    "timeout": 60,
    "regionSize": 50000,
    "skills": "System Programmer",
    "output": null,
    "scriptParameters": "para1",
    "isRestStep": false,
    "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
    "outputSysoutDD": null,
    "variable-specifications": [
      {
        "scope": "instance",
        "name": "st_group",
        "required": true
      },
      {
        "scope": "instance",
        "name": "st_user",
        "required": true
      },
      {
        "scope": "instance",
        "name": "procNameVariable",
        "required": true
      }
    ],
    "name": "TSO-UNIX-shell_Execution",
    "successPattern": "success.*",
    "saveAsDataset": null
  },
  {
    "template": "/* rexx */\nparse arg arg1\n\nSAY \"this is a sample to submit UNIX REXX script to run immediately\"\n\nSAY \"the first parameter is :\" arg1\n\nSAY ${instance-st_user}\n\nSAY \"prefix:st_group =\" SYS123\n\nSAY \"prefix:st_user =\" USERS\n\nSAY \"This symbol is used to indicate failed\"\n\n",
    "instructions": "This step outputs some variables and prints a few words.\n",
    "autoEnable": "false",
    "maxLrecl": 1024,
    "submitAs": "TSO-UNIX-REXX",
    "failedPattern": [
      "failed.*"
    ],
    "description": "In this step, you submit an inline UNIX REXX exec for immediate processing \n\t\nton the host system. In this example, the step is expected to fail.\n\t\t",
    "weight": 1,
    "outputVariablesPrefix": "prefix:",
    "optional": false,
    "procName": "${instance-procNameVariable}",
    "title": "A step that runs a UNIX REXX exec program.",
    "timeout": 60,
    "regionSize": 50000,
    "skills": "System Programmer",
  }
]

```

Figure 479. Sample response from a get workflow definition request (Part 2 of 3)

```

        "output": null,
        "scriptParameters": "para1",
        "isRestStep": false,
        "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
        "outputSysoutDD": null,
        "variable-specifications": [
            {
                "scope": "instance",
                "name": "st_group",
                "required": true
            },
            {
                "scope": "instance",
                "name": "st_user",
                "required": true
            },
            {
                "scope": "instance",
                "name": "procNameVariable",
                "required": true
            }
        ],
        "name": "TSO-UNIX-REXX_Execution",
        "successPattern": "success.*",
        "saveAsDataset": null
    },
    {
        "template": "/* rexx */\nparse arg arg1\nSAY \"this is a sample to submit TSO REXX script to run immediately\"\nSAY \"the first parameter is :\" arg1\nSAY ${instance-st_user}\nSAY \"prefix:st_group =\" SYS123\nSAY \"prefix:st_user =\" USERS\nSAY \"This execution will meets timeout.\"\n",
        "instructions": "This step outputs some variables and prints a few words.\n",
        "autoEnable": "false",
        "maxLrecl": 1024,
        "submitAs": "TSO-REXX",
        "failedPattern": [
            "failed.*"
        ],
        "description": "In this step, you submit an inline REXX exec for immediate processing \n\t\tton the host system.\nIn this example, the processing is ended by a time-out condition.\n\t\t",
        "weight": 1,
        "outputVariablesPrefix": "prefix:",
        "optional": false,
        "procName": "${instance-procNameVariable}",
        "title": "A step that runs a REXX exec program.",
        "timeout": 60,
        "regionSize": 50000,
        "skills": "System Programmer",
        "output": null,
        "scriptParameters": "para1",
        "isRestStep": false,
        "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
        "outputSysoutDD": null,
        "variable-specifications": [
            {
                "scope": "instance",
                "name": "st_group",
                "required": true
            },
            {
                "scope": "instance",
                "name": "st_user",
                "required": true
            },
            {
                "scope": "instance",
                "name": "procNameVariable",
                "required": true
            }
        ],
        "name": "TSO-TSO-REXX_Execution",
        "successPattern": "success.*",
        "saveAsDataset": null
    }
],
    {
        "productName": "Product ABC",
        "globalVariableGroup": null,
        "containsParallelSteps": false,
        "workflowDefinitionFileMD5Value": "5c5dd66eb3ca3cd1c578ccf323d57cc0",
        "isCallable": null,
        "productVersion": "Version 1",
        "jobsOutputDirectory": null,
        "vendor": "IBM",
        "scope": "none",
        "workflowVersion": "1.0",
        "category": "configuration",
        "workflowID": "programExecutionSample"
    }
}

```

Figure 480. Sample response from a get workflow definition request (Part 3 of 3)

Archive a workflow instance

You can use this operation to archive a z/OSMF workflow instance on a z/OS system.

HTTP method and URI path

```
POST /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/archive
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- *<workflowKey>* identifies the workflow to be archived.

Query parameters

None.

Description

This operation archives a workflow instance, which is identified by the workflow key that is specified in the request URI.

You can archive any workflows that are completed or that you no longer need. Doing so removes the workflow from the Workflows table in the Workflows task and places it in an archive for your reference. An archived workflow is no longer active, but its information can be viewed by you at any time. You cannot undo this action.

After you archive a workflow, you can list it or delete it, or retrieve its properties.

When you no longer want to retain an archived workflow, you can delete it permanently from z/OSMF.

To be archived, a workflow must be in one of the following states:

- In-progress
- Complete
- Canceled.

It is not possible to archive a workflow while it is involved in a workflow activity. Specifically, you cannot archive a workflow when it:

- Is locked for an update operation
- Contains an automated step that is running
- Is called for processing by another workflow.

To do so, you must allow the processing to complete first.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the archival of the workflow. The URI path for the workflow is provided in the Location response header and a response body is provided, as described in the [“Response content” on page 1079](#).

Request content

None.

Authorization requirements

See [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 1079](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code and the associated error message.

Table 564. HTTP error response codes for an archive workflow request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an incorrect parameter, such as an incorrect workflow key.
HTTP 403 Forbidden	The requestor user ID is not permitted to archive the workflow.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	Request cannot be performed because the specified workflow has a status that makes the workflow ineligible to be archived.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the service returns the response body, which contains a JSON object with the workflow key. [Table 565 on page 1079](#) describes the contents of the response body.

Table 565. Response from an archive workflow request		
Field name	Type	Description
workflowKey	String	Workflow key. A string value, generated by z/OSMF, which is used to uniquely identify the archived workflow instance.

Example HTTP interaction

In [Figure 481 on page 1079](#), a request is submitted to archive the workflow that is identified by the workflow key 2535b19e-a8c3-4a52-9d77-e30bb920f912.

```
POST /zosmf/workflow/rest/1.0/workflows/2535b19e-a8c3-4a52-9d77-e30bb920f912/operations/archive
Host: zosmf1.yourco.com
Connection: close
```

Figure 481. Sample request to archive a workflow

A sample response is shown in [Figure 482 on page 1079](#).

```
HTTP/1.1 201 Created
{
  "workflowKey": "2535b19e-a8c3-4a52-9d77-e30bb920f912"
}
```

Figure 482. Sample response from an archive workflow request

List the archived workflows for a system

You can use this operation to list the archived z/OSMF workflows for a system or sysplex.

HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/archivedworkflows
```

In this request, the URI path variable *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.

Query parameters

Optionally, your request can include one or more of the following query parameters to filter the results:

Orderby

To sort the returned instances by time, specify either of the following values:

"desc":

From the newest to the oldest

"asc":

From the oldest to the newest

View

An string type to select the list instances by view:

"user":

Return the archived workflow instances that are owned by the user, up to a maximum of 200 workflow instances. This value is the default.

"domain":

For archived provisioning workflows, return the instances that the user is authorized to view. The user must be a domain owner. The results are grouped by domain, with up to 200 instances per domain.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark ('?') to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).

Description

This operation retrieves a list of archived workflows that you are authorized to view.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 567 on page 1081](#).

Authorization requirements

See [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 566. HTTP error response codes for a get archived workflow properties request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an error, such as an incorrect query parameter.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body contains one property, which is called archived workflows. This property is an array of workflow-info objects. Table 567 on page 1081 lists the fields in the workflow-info object. If no workflows match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 567. List archived workflows request: Format of the workflow-info object		
Field name	Type	Description
workflowName	String	Descriptive name for the workflow.
workflowKey	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
archivedInstanceURI	String	Workflow instance URI path, which you can use to retrieve information about the archived workflow.

Example HTTP interaction

In the following example, the GET method is used to list the archived workflows on a system. Here, the query parameter `?orderBy=desc` is included to order the results in descending order.

```
GET /zosmf/workflow/rest/1.0/archivedworkflows?orderBy=desc HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 483. Sample request to list archived workflows

For a successful request, the HTTP response includes a JSON document that contains the requested information. In the following example, three archived workflows were found for the requestor user ID.

```

HTTP/1.1 200 OK
content-length: 464
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8

HTTP/1.1 200 OK
{
  "archivedWorkflows": [
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_5",
      "workflowKey": "2535b19e-a8c3-4a52-9d77-e30bb920f912",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
/2535b19e-a8c3-4a52-9d77-e30bb920f912"
    },
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_0",
      "workflowKey": "8f0f572a-0eb5-493e-91b2-3d549374e07d",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
/8f0f572a-0eb5-493e-91b2-3d549374e07d"
    },
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_5",
      "workflowKey": "1aead54d-3507-4473-9cda-e4fd25eb21b8",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
/1aead54d-3507-4473-9cda-e4fd25eb21b8"
    }
  ]
}

```

Figure 484. Sample response from a list archived workflows request

Get the properties of an archived workflow

You can use this operation to retrieve the properties of an archived z/OSMF workflow.

HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the archived workflow to be queried.

Query parameters

You can specify the following query parameter on this request:

returnData

This optional query parameter is used to request information about the workflow steps and variables. Include one or both of the following attributes on the `returnData` parameter:

steps

Returns an array of step-info objects; one object for each step in the workflow. [Table 571 on page 1088](#) lists the fields in the step-info JSON object.

variables

Returns an array of variable-info objects; one object for each variable that is referenced in the workflow. [Table 572 on page 1095](#) lists the fields in the variable-info JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:


```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

Description

This operation retrieves the properties of an archived z/OSMF workflow. You can optionally expand the returned information through the specification of query parameters. On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 569 on page 1083](#).

For the format of this information, see the JSON objects that are described in [Table 571 on page 1088](#) and [Table 572 on page 1095](#).

Authorization requirements

See [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 569 on page 1083](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 568. HTTP error response codes for a get archived workflow properties request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an incorrect parameter, such as an incorrect workflow key.
HTTP 403 Forbidden	The requestor user ID is not permitted to retrieve the workflow properties.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 569 on page 1083](#) lists the fields in the JSON object.

Table 569. JSON object that is returned to a get archived workflow properties request		
Field name	Type	Description
workflowName	String	Descriptive name for the workflow.
workflowKey	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
workflowDescription	String	Description of the workflow.
workflowID	String	Workflow ID. A short, arbitrary value that identifies the workflow.
workflowVersion	String	Version of the workflow definition file.

Table 569. JSON object that is returned to a get archived workflow properties request (continued)

Field name	Type	Description
workflowDefinitionFileMD5Value	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
vendor	String	Name of the vendor that provided the workflow definition file.
owner	String	User ID of the workflow owner.
system	String	Full name of the z/OS system on which the workflow is to be performed. This value is in the format <i>sysplex.sysname</i> .
category	String	Category of the workflow, which is either general or configuration.
productID	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
productName	String	Name of the product or component that is being configured through the workflow.
productVersion	String	Version and release of the product or component that is configured through the workflow.
percentComplete	Integer	Percentage of the workflow that is completed. z/OSMF calculates this value based on the number of steps in the workflow and the relative weighting value of each step.
isCallable	Boolean	Indicates whether a workflow is eligible to be called by another workflow. For more information, see “Callable workflows” on page 1111.
containsParallelSteps	Boolean	For a parallel-steps workflow, this property is <code>true</code> . If so, the automation ready steps can be run in parallel (concurrently), thus possibly completing more quickly. Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
scope	String	Restricts a workflow to one instance only. The scope attribute can be set to <code>system</code> , <code>sysplex</code> , or <code>none</code> . For more information, see “Setting the workflow scope” on page 1111.
statusName	String	Indicates the current workflow status, which is <i>archived</i> .
deleteCompletedJobs	Boolean	For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully, as follows: <ul style="list-style-type: none"> <code>false</code> means that the job is retained on the JES spool until it is removed by a user or automated process. <code>true</code> means that the job is deleted from the JES spool after it completes or fails.

Table 569. JSON object that is returned to a get archived workflow properties request (continued)

Field name	Type	Description
automationStatus	Object	<p>An automation-info object that contains details about the most recent start automation request for the workflow. The content of this property depends on the following factors:</p> <ul style="list-style-type: none"> • If no automation was performed for the workflow, this property is null. • If automation processing is still in progress, this property indicates the step that is being processed. • If automation was restarted after it was stopped, this property indicates the status of the current start automation request. • If automation is stopped and the workflow status is complete, this property indicates that automation is completed. • If automation is stopped and the workflow status is not complete, this property identifies the step that is most closely related to the reason why automation was stopped. <p>Notes about parallel-step workflows:</p> <ul style="list-style-type: none"> – When a parallel-steps workflow is started, all of its automation ready steps are processed until they complete or fail, or automation is stopped. Failure of a step does not stop automation processing for other automation ready steps in the workflow. – In a parallel-steps workflow: <ul style="list-style-type: none"> - The automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. - If automation is currently stopped and the workflow is not yet complete, this property identifies the first uncompleted step that was returned to the Get Properties request. <p>Table 570 on page 1087 lists the fields in the automation-info object.</p>
jobsOutputDirectory	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
autoDeleteOnCompletion	Boolean	<p>Specifies whether the workflow is automatically deleted from the system after it completes successfully, as follows:</p> <ul style="list-style-type: none"> • <code>false</code> means that the workflow is retained after it is complete, until it is removed by a user. A complete workflow is one in which all of its steps are marked complete or skipped. • <code>true</code> means that the workflow is automatically deleted from the system after it completes. As a result, the workflow is removed from the Workflows table in the Workflows task user interface.

Table 569. JSON object that is returned to a get archived workflow properties request (continued)

Field name	Type	Description
access	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in “Workflow access type” on page 1023 . The following values are valid: <ul style="list-style-type: none"> • Public • Restricted • Private
archivedTime	String	Date and time on the system when the workflow was archived. The date and time is in Greenwich Mean Time (GMT).
accountInfo	String	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
jobStatement	String	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 to 8 characters.
steps	Array of objects	Array of one or more step-info objects that contain details about each of the steps in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> . The content of this array depends on what the requestor is permitted to see. For more information, see “Description” on page 1083 . Table 571 on page 1088 lists the fields in the step-info object.
variables	Array of objects	Array of one or more variable-info objects that contain details about the variables that are used in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> . The content of this array depends on what the requestor is permitted to see. For more information, see “Description” on page 1083 . Table 573 on page 1095 lists the fields in the variable-info object.

Format of the automation-info object

[Table 570 on page 1087](#) lists the fields in the automation-info JSON object.

Table 570. Get Archived Workflow Properties request: Format of the automation-info object		
Field name	Type	Description
startUser	String	User ID of the user who initiated the automation processing.
startedTime	Timestamp	Time that automation processing started. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
stoppedTime	Timestamp	Time that automation processing stopped. If automation is still in progress, this property is set to null. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
currentStepName	String	Depending on the current phase of automation processing, this property contains one of the following values: <ul style="list-style-type: none"> Name of the step that is being processed automatically. Name of the step that caused automation to stop. For a workflow that uses parallel processing (a <i>parallel-steps workflow</i>) this value is the name of the first step that is incomplete. If automation is stopped and the workflow status is complete, this property is set to null.
currentStepNumber	String	The step number. If automation is stopped and the workflow status is complete, this property is set to null.
currentStepTitle	String	Step title. If automation is stopped and the workflow status is complete, this property is set to null.
messageID	String	Message identifier for the accompanying message. If automation is still in progress, this property is set to null.
messageText	String	Message text that describes the reason that automation is stopped. If automation is still in progress, this property is set to null.

Format of the step-info object

Table 571 on page 1088 lists the fields in the step-info JSON object. Not all of the properties are returned for every step. Some properties are returned or omitted depending on the step type, as noted in the **When returned** column. This information in this column indicates whether valid data is returned for the step, as follows:

All step types

Properties that are returned for all step types.

Calling steps

Properties that are returned for a step that calls another workflow for execution.

Template steps

Properties that are returned for a step that runs a program, such as a batch job, REXX exec, or UNIX shell script.

REST steps

Properties that are returned for a step that issues a REST API request, such as a GET or PUT request.

With regard to returned data, a template step and a REST step are mutually exclusive. The returned information for a template step does not include the properties for a REST step. Likewise, the returned information for a REST step does not include the properties for a template step. To help you identify which steps are REST steps, the step-info object includes the `isRestStep` property, set to true or false.

Table 571. Get Archived Workflow Properties request: Format of the step-info object			
Field name	Type	When returned	Description
name	String	All step types	Name of the step.
actualStatusCode	String	REST steps only	The actual HTTP status code received from the REST API request. To obtain this value, map it to a workflow variable.
assignees	String	Calling steps and template steps	Step assignees; one or more user IDs that are assigned to the step. Multiple items are separated by commas.
autoEnable	Boolean	All step types	Indicates whether the step can be performed automatically when all prerequisite steps are completed, and no user inputs are required.
calledInstanceURI	String	Calling steps only	For a step that calls another workflow for execution, this property contains the URI path of the called workflow instance. You can use this value to retrieve information about the called workflow. This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
calledWorkflowID	String	Calling steps only	This property contains the workflow ID of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property <code>calledWorkflowMD5</code> is specified.
calledWorkflowVersion	String	Calling steps only	This property contains the workflow version of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property: <ul style="list-style-type: none"> • Is null when the property <code>calledWorkflowMD5</code> is specified • Might be null when the property <code>calledWorkflowID</code> is specified.
calledWorkflowMD5	String	Calling steps only	This property contains the 128-bit hash value of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property <code>calledWorkflowID</code> is specified.
calledWorkflowDescription	String	Calling steps only	This property contains a description of the workflow to be called, from the point of view of the calling workflow.

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
calledWorkflowDefinitionFile	String	Calling steps only	This property contains the name of the workflow definition file that will be used to create a new workflow if an existing instance is not found when this step is performed. This property might be null.
description	String	All step types	Step description.
expectedStatusCode	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
failedPattern	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
hasCalledWorkflow	Boolean	Calling steps and template steps	Indicates whether this step calls another workflow (true or false). If true, this step is a "calling" step, that is, it calls another workflow for execution. If false, it is a template step. This property is returned only when steps=null, which indicates a leaf step.
hostname	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
httpMethod	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> • GET • PUT • POST • DELETE
instructions	String	Template steps only	Detailed instructions on what the user must do to perform the step.
instructionsSub	Boolean	Template steps only	Indicates whether the step instructions contain variables (true or false).
isConditionStep	Boolean	Calling steps and template steps	Indicates whether this step is a conditional step (true or false).

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
isRestStep	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> • actualStatusCode • expectedStatusCode • hostname • hostnameSub • httpMethod • port • portSub • queryParameters • queryParametersSub • requestBody • requestBodySub • schemeName • schemeNameSub • uriPath • uriPathSub <p>The following step properties are not applicable for a REST step and thus, are omitted from the output:</p> <ul style="list-style-type: none"> • template • templateSub • output • outputSub • saveAsDataset • saveAsDatasetSub • saveAsUnixFile • saveAsUnixFileSub • submitAs • maxLrecl • returnCode
maxLrecl	Integer	Template steps only	For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.
optional	Boolean	All step types	Indicates whether the step is optional (true or false).

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
output	String	Template steps only	Indicates the name of the output file produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
outputSub	Boolean	Template steps only	Indicates whether the output file name contains variable substitution (true or false).
outputVariablesPrefix	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
owner	String	Calling steps and template steps	User ID of the step owner.
port	String	REST steps only	Port number that is associated with the REST request.
portSub	Boolean	REST steps only	Indicates whether the port number contains variable substitution (true or false).
prereqStep	Array of strings	All step types	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
procName	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.
queryParameters	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
queryParametersSub	Boolean	REST steps only	This property indicates whether the query parameters contain variable substitution (true or false). Otherwise, this property is null.
regionSize	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
requestBody	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
requestBodySub	Boolean	REST steps only	This property indicates whether the request body variable substitution (true or false). Otherwise, this property is null.
returnCode	String	Template steps only	For a step that submits a job to run, this property indicates the return code that was returned when the job was submitted.
saveAsDataset	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.
saveAsDatasetSub	Boolean	Template steps only	Indicates whether the data set name contains variable substitution (true or false).
saveAsUnixFile	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
saveAsUnixFileSub	Boolean	Template steps only	Indicates whether the UNIX file name contains variable substitution (true or false).
schemeName	String	REST steps only	The scheme name that is used for the REST request. For example: <code>http</code> .
schemeNameSub	Boolean	REST steps only	Indicates whether the scheme name contains variable substitution (true or false).
scriptParameters	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
skills	String	Calling steps and template steps	The type of skills that are required to perform the step.

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
state	String	All step types	<p>State of the step. One of the following status indicators is displayed:</p> <ul style="list-style-type: none"> • Unassigned. The step is in the <i>Unassigned</i> state; no users or groups are assigned to the step. • Assigned. Users or groups are assigned to the step, but no user accepted ownership of the step. • Not Ready. A user accepted ownership of the step, however, a prerequisite step must be completed or a conditional dependency must be satisfied before the step can be performed. • Ready. The step is ready to be performed; all prerequisite steps and conditional dependencies are satisfied. • In Progress. The step is in progress. For a parent step, a state of <i>In Progress</i> means that at least one of the child steps is started, but is not yet complete, overridden, or skipped. For a leaf step, a state of <i>In Progress</i> means that the step is started, but is not yet complete, overridden, or skipped. • Submitted. The step included a job, which the step owner submitted. • Complete. The step was completed. • Skipped. The step was bypassed by the step assignee. • Complete (Override). The step was marked complete, but the work was performed outside of the Workflows task. • Failed. The step included a job that was submitted by the step owner. However, the job failed to complete successfully. • Conflicts. The step created an output file for use by a subsequent step. However, values in that file conflict with existing instance or global variables. • Condition Not Satisfied. The step is a conditional step, and the condition is not satisfied.
stepNumber	String	All step types	The step number. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.
steps	Array of objects	All step types	<p>For a parent step, this is a nested array of step-info objects. For a leaf step, this property is null.</p> <p>Check this property first before you check the other, non-common step properties. A non-null value here means that the calling step properties are omitted, as are the template step properties and the REST step properties.</p>

Table 571. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
submitAs	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The possible values are the following: <ul style="list-style-type: none"> • "JCL " • "TSO-REXX" • "shell-JCL " • "TSO-REXX-JCL " • "TSO-UNIX-REXX" • "TSO-UNIX-shell "
successPattern	String	Template steps only	Regular expression that is returned for a successful program execution.
template	String	Template steps only	Indicates the template that is used to run a program or batch job (inline or external file).
templateSub	Boolean	Template steps only	Indicates whether template contains variable substitution (true or false). The default is false.
timeout	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.
title	String	All step types	Step title.
uriPath	String	REST steps only	The URI path to use for the REST request.
uriPathSub	Boolean	REST steps only	Indicates whether the URI path contains variable substitution (true or false).
userDefined	Boolean	All step types	Indicates whether the step was added manually to the workflow (true or false). If true, the step was added by the workflow owner, using the Update Workflow Steps action in the Workflows table. If false, the step was defined in the workflow definition that was used to create the workflow.
variable-references	Array of objects	Template steps only	An array of variable-reference objects, the format of which is described in Table 572 on page 1095 .
weight	Integer	Calling steps and template steps	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

Format of the variable-reference object

Table 572 on page 1095 lists the fields in the variable-reference JSON object.

Table 572. Get Archived Workflow Properties request: Format of the variable-reference object		
Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.

Format of the variable-info object

Table 573 on page 1095 lists the fields in the variable-info JSON object.

Table 573. Get Archived Workflow Properties request: Format of the variable-info object		
Field name	Type	Description
name	String	Name of the variable.
scope	String	Variable scope, which is either instance or global.
type	String	Type of variable, which is one of the following values: <ul style="list-style-type: none">• boolean• string• number• date• time• array
value	String	Variable value.
visibility	String	Public or private.

Example HTTP interaction

In the following example, the GET method is used to retrieve information about an archived workflow. The workflow is uniquely identified by the workflow key, which is represented by the following string value: 2535b19e-a8c3-4a52-9d77-e30bb920f912.

```
GET /zosmf/workflow/rest/1.0/archivedworkflows/2535b19e-a8c3-4a52-9d77-e30bb920f912
HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 485. Sample request to get archived workflow properties

An example of the response is shown in [Figure 486 on page 1096](#).

```

HTTP/1.1 200 OK{
  "access": "Public",
  "productID": "ABC123",
  "jobStatement": null,
  "deleteCompletedJobs": false,
  "productName": "Product ABC",
  "globalVariableGroup": null,
  "productVersion": "Version 1",
  "jobsOutputDirectory": null,
  "vendor": "IBM",
  "archivedTime": "2020-03-04 03:18:36",
  "scope": "none",
  "statusName": "archived",
  "workflowID": "programExecutionSample",
  "owner": "zosmfad",
  "accountInfo": null,
  "isInstanceVariableWithoutPrefix": false,
  "workflowName": "testProgramExecutionSample",
  "automationStatus": null,
  "autoDeleteOnCompletion": false,
  "percentComplete": 0,
  "workflowDescription": "Sample that demonstrates how to run an executable program from a step.\n\t",
  "containsParallelSteps": false,
  "workflowDefinitionFileMD5Value": "5c5dd66eb3ca3cd1c578ccf323d57cc0",
  "isCallable": null,
  "system": "PLEX1.SY1",
  "workflowKey": "7a2263a7-7c91-40b4-8892-2a4342a222c3",
  "workflowVersion": "1.0",
  "category": "configuration"
}

```

Figure 486. Sample response from a get archived workflow properties request

Delete an archived workflow

You can use this operation to remove an archived z/OSMF workflow from a z/OS system.

HTTP method and URI path

```
DELETE /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the archived workflow to be deleted.

Query parameters

None.

Description

This operation is used to delete an archived workflow from z/OSMF, including any notes that accompany the workflow and its steps, and the history log for the workflow.

Request content

None.

Authorization requirements

For a general workflow or configuration workflow, the ability to delete the workflow is limited to the current workflow owner and members of the z/OSMF workflow administrators group. For a provisioning

workflow, the domain administrator is also able to delete a workflow. A delete request from another user is rejected with the HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 1021](#).

HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 574. HTTP error response codes for a delete archived workflow request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an incorrect parameter, such as an incorrect workflow key.
HTTP 403 Forbidden	The requestor user ID is not permitted to delete the workflow properties.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.

Other standard status codes can be returned, as described in [“HTTP status codes” on page 1022](#).

Response content

None.

Example HTTP interaction

In the following example, the DELETE method is used to delete an archived workflow. The workflow is identified by the workflow key, which is the following string value: 7c4bac42-16a3-4af5-a5b9-263e60b280a4.

```
DELETE /zosmf/workflow/rest/1.0/archivedworkflows/7c4bac42-16a3-4af5-a5b9-263e60b280a4 HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 487. Sample request to delete an archived workflow

For a successful request, the HTTP response 204 is returned.

```
HTTP/1.1 204 No Content
```

Figure 488. Sample response from a delete archived workflow request

Chapter 2. Creating workflow definitions for z/OS

This information describes how to create workflow definitions that can be used with the Workflows task of z/OSMF. Included is an introduction to workflows concepts and a description of the basic elements of a workflow definition.

What is a z/OSMF workflow?

Generally, a *workflow* guides you through the complete set of steps that are needed to accomplish a goal, and, when dependencies exist, controls the sequence for performing those steps. In this way, a workflow can help to ensure that the steps are performed in the correct order, and prerequisites and dependences are identified clearly along the way. Conceptually, a workflow encompasses both the work to be performed and its performers. By identifying the individual steps to be performed, a workflow allows for the steps to be divided among different areas of an organization, and different members of a team. Using a workflow, a project owner can delegate specific items to the team members best suited to carrying out particular tasks.

In z/OSMF, a workflow is a guided set of steps that help you perform an activity on z/OS, such as configuring a software product or component, managing a z/OS resource or structure, or simplifying some relatively complex operation. To support these activities, a workflow can be designed to perform a wide variety of operations, such as starting z/OS subsystems, submitting jobs and scripts, and invoking TSO/E functions in batch (assuming that the workflow user is properly authorized).

In short, a z/OSMF workflow:

- Is based on a structured set of steps that are designed by a *workflow author*.
- Is described to z/OSMF through a *workflow definition*. A z/OS organization can write its own workflow definitions or obtain definitions from a third-party source (a *workflow provider*). z/OSMF includes samples for the workflow authors to reference when they create workflow definitions.
- Is created when a user imports a workflow definition into the z/OSMF Workflows task.
- Identifies steps to be performed and allows for these steps to be divided among different areas of an organization, which helps to facilitate user activities on z/OS.
- Contains one or more steps that guide the user through some action to be performed. Steps might consist of manual instructions for performing the steps, or might include some form of guided assistance, such as submitting a batch job, running a REXX script or a shell script, or creating files, based on user inputs.

In z/OSMF, the Workflows task allows a z/OS installation to create and manage workflows for performing activities on the z/OS system. The user who is responsible for the workflow and ensuring that it gets completed is the *workflow owner*. The workflow owner assigns workflow steps to users, making them *assignees* of the step. The user who accepts ownership of a step becomes the *step owner*.

In the Workflows task:

- The **Workflows** page displays the existing workflows for an installation, and provides the control point for creating and managing workflows.
- The **Steps** page displays the steps in a workflow, and provides the control point for managing the steps. From this page, you can select actions for the steps, such as assigning steps, changing ownership of steps, and performing steps.

The following topics provide more details on workflows, workflow steps, and the process of creating workflow definition files for use with the Workflows task of z/OSMF:

- [“Terms you should know” on page 1100](#)
- [“The Workflows task schema” on page 1102](#)
- [“Creating the workflow definition file” on page 1102](#)
- [“Defining steps for your workflow” on page 1117](#)

- [“Defining variables for your workflow” on page 1154](#)
- [“Workflow XML reference” on page 1173.](#)

Terms you should know

Workflow authors should be familiar with the following terms.

Workflow

1. An activity that is associated with the z/OS system, such as configuring a component or product. **2.** The instantiation of a workflow in z/OSMF, based on a workflow definition. A workflow consists of one or more units of work to be performed on the z/OS system, as described by the workflow definition. A workflow is created when the Workflows task is used to create an instance of a workflow from a supplied workflow definition file.

Workflows task

The task in the z/OSMF desktop that allows users to interact with workflows on z/OS.

Workflow category

A classification of the activities to be performed in the workflow. In z/OSMF, a workflow that is used to configure system software is classified as a *Configuration* workflow. A workflow that is used to provision system software is classified as a *Provisioning* workflow. All other workflows are classified as *General* workflows.

Workflow definition

The logical structure of a workflow, represented as a series of one or more steps. The workflow definition identifies the various system objects and actions that constitute activities on z/OS and the rules for performing those activities. The workflow definition includes all of the information that is specified in, or referenced by, the primary XML file (the *workflow definition file*) and possibly other files that are included by the workflow definition file. This content typically includes information about the workflow (such as name and version), step definitions, variable definitions, file templates, and bundle files.

Workflow definition file

The primary XML file for a workflow definition. A workflow is stored in z/OSMF when its workflow definition file, and optionally, a workflow variable input file, is imported into the Workflows task.

Workflow variable input file

An optional file that supplies default values for one or more of the input variables that are defined in the workflow definition file. The workflow variable input file is specified as input when the workflow definition is imported into the Workflows task. Typically, a workflow provider might supply a workflow variable input file to save users from having to manually enter inputs when they perform a workflow.

Job output file

For a workflow that runs a batch job, the workflow author can specify that job spool files are to be saved in a user-specified location (a UNIX directory). The file that is used to save the contents of a job spool file is referred to as a *job output file*.

Output properties file

A file that is created at the completion of a step. Typically, the output properties file holds the results of a batch job, shell script, or REXX exec program, as determined by the workflow author. The output properties file can be used by other steps or workflow instances. In practice, a step might submit a batch job to create some z/OS related parameters, which are then used by a subsequent step, thus saving the Workflows task user from having to enter the parameter values manually.

Array variable

A variable definition that can be used to represent a list of values or name-value pairs.

Global variable

A variable definition that is available to all workflow instances. The Workflows task saves global variables in a repository that is called the global variables pool.

Note: Global variables are deprecated, as of z/OS V2R3. IBM recommends that you use instance variables or system variables, instead. Global variables might not be supported in a future release.

Instance variable

A variable definition that is available only to instances of a particular workflow.

Predefined variable

A variable definition that can be used for string substitution in the current step only.

System variable

A variable definition that is created through the z/OSMF system variable services, which is a REST application programming interface. For more information, see [“z/OS system variable services” on page 802](#).

Workflow author

The person, typically a programmer, who creates the workflow definition by using the XML tagging language.

Workflow owner

The user who is given ownership of the workflow through the Workflows task. The workflow owner is responsible for delegating the work in the workflow to users to perform (the step assignees).

Workflow provider

The source of the workflow definition file, typically IBM, or a software vendor.

Step

A single, logical unit of work in a workflow. Consider each step to describe a specific activity to be performed on the system. A step is available to be performed when the workflow owner assigns the step to a user through the Workflows task, and the user accepts ownership of the step.

Step owner

The user who accepts ownership of a step and therefore responsibility for performing the step.

Automation processing

The processing of a workflow that contains one or more automated steps. A workflow that is comprised entirely of automated steps can complete with little or no user intervention. When automation processing is started on the workflow, the workflow runs to completion or until it is stopped by another condition, such as a user request or an error.

Automated step

A step can be designed to run automatically (without user interaction) when it is in Ready state. Such a step is referred to as an *automated step*. A workflow that is comprised entirely of automated steps can complete with little or no user intervention.

Batch execution step

A template step that runs an executable program as a batch job, such as a JCL job, a REXX exec, or a UNIX shell script. Contrast with an *immediate execution step*, which is a template step that runs a program in real time.

Conditional step

A step that can be performed when a logical condition is satisfied on the z/OS system or in the Workflows task. For example, a conditional step might become eligible to be performed if a job that is run by another step ends with a particular return code. A conditional step remains unavailable to be performed as long as the condition is not satisfied.

Called workflow

A workflow that is started by another workflow for execution. Conceptually, a called workflow is a step in the workflow that calls it (the calling workflow).

Feedback step

A step that includes a feedback form with questions for the step owner to answer at the conclusion of a step.

Immediate execution step

A template step that runs an executable program in real time, such as a REXX exec or UNIX shell script. Contrast with a *batch execution step*, which is a template step that runs a program as a batch job.

REST step

A step that issues a REST API request, such as a GET or PUT request.

Template step

A step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script. On completion, the results can be made available to other steps, in the form of variables or an output properties file. Depending on how the program is processed, a template step is either of the following:

- *Immediate execution step*, which runs a program in real time
- *Batch execution step*, which runs a program as a batch job.

The Workflows task schema

A valid workflow definition file is one that follows the XML syntax and also conforms to the rules of the Workflows task schema.

The Workflows task schema is supplied with z/OSMF in the following location:

```
/usr/lpp/zosmf/workflow/schemas/workflow_v1.xsd
```

The schema file is UTF-8 encoded.

If you are developing a workflow definition file, you require access to the schema, and therefore access to the z/OS system that is running z/OSMF.

Creating the workflow definition file

This topic describes the elements that comprise the workflow definition file.

A workflow is defined through a workflow definition file, which consists of one or more XML files and other types of files. Depending on the workflow design, a workflow might consist of just a single workflow definition file, or it might have a primary XML file that references one or more subordinate XML files, XML fragments, and external files. This document uses the term *workflow definition file* to refer collectively to all of the files that define a given workflow.

As a workflow author, you can create a workflow definition file in XML, in accordance with the schema that is supplied with the Workflows task of z/OSMF. The schema defines the required and optional properties (XML elements and attributes) of a workflow and imposes constraints on the order in which the elements are specified, and on the values that can be specified for each element and attribute.

It is assumed that workflow authors are familiar with the XML specification and coding practices. The following references provide additional helpful information:

- The World Wide Web Consortium (W3C) XML Technology page: <http://www.w3.org/standards/xml/>
- XML Core Working Group Public Page: <http://www.w3.org/XML/>

Besides XML files, a workflow definition might include external files. That is, apart from XML fragments, the workflow definition can refer to translated text files and velocity template files. These files must be read-accessible by the user who is creating (importing) the workflow in the z/OSMF Workflows task.

You can provide the workflow definition file and any associated files in either a z/OS UNIX file or a z/OS data set. For a z/OS data set, use a sequential data set or a member of a partitioned data set (PDS).

For an example of how you can refer to an external file or fragment from a workflow definition file, see [“Defining entities for a workflow” on page 1107](#).

Workflow Editor task in z/OSMF

To help you with creating and editing a workflow definition, z/OSMF includes an editor for workflows. You can use the Workflow Editor task to view, create, and modify workflow definitions. The Workflow Editor provides a visual framework for working with the elements of a workflow definition—the steps, variables, and workflow metadata.

The Workflow Editor task:

- Presents the details of a workflow definition in a graphical user interface (GUI).

- Provides easy-to-use options for viewing, creating, and modifying a workflow definition.

Using the Workflow Editor task, you can:

- Select an existing workflow definition file for editing. Or, have the Workflow Editor create a starter workflow with which you can begin working.
- View details about the different sections of a workflow definition—the metadata, steps, and variables.
- Modify the workflow information, steps, and variables sections of the definition, including adding and deleting steps and variable definitions.
- Edit the workflow variable input file, if one is available for use with the workflow definition.
- Overwrite the workflow definition with your changes.

To get started with the Workflow Editor task, in the z/OSMF desktop, select **Workflow Editor**.

The workflow definition file must be valid XML, otherwise it cannot be opened in the editor.

If the workflow definition file resides in a z/OS UNIX file path, ensure that the file permissions are set as follows:

- A workflow definition file must allow read and write access for the "user" bit, but requires only read access for the "other" bit (that is, file permissions of at least 604).
- Any external files that are referenced by the workflow definition must have file permissions of at least 604.
- The directory location for these files requires file permissions of at least 505.

If the workflow definition file resides in a data set, ensure that your user ID has write access to the data set. Also, the z/OSMF server user ID, which is IZUSVR by default, requires read access to the data set. To verify that the server has access, contact your z/OSMF administrator.

More information about the Workflow Editor is provided in the online help.

Structure of a workflow definition file

Structurally, a workflow definition file is comprised of several sections, as follows:

- **Document declaration statements** that are not directly related to workflow content. These statements are required at the beginning of every workflow XML file, and are described as follows:

XML processing instruction

The primary XML file must start with a processing instruction (in column 1 of line 1) for the XML processor. This instruction defines the version of XML used and the encoding of the file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Document type definition (DTD)

You can optionally use a DTD to define entities (variables) in the workflow. Using a DTD with workflows is optional because z/OSMF uses the default XML schema to validate the contents of a workflow file, rather than the DTD.

Workflow root element

The workflow root element is the container for the main content of a workflow.

- **Workflow metadata**, which contains information about the workflow itself. For more information, see [“Specifying the workflow metadata” on page 1109](#).
- Optionally, a **manifest** of external files that contain translated text for the various user interface elements (widgets) that are displayed for the workflow. The Workflows task of z/OSMF can display text for widgets in the language defined for the browser locale, if you supply the translated text in a properties-based resource bundle file. By default, Workflows task displays the text in whatever language is used in the workflow file. Thus, it is recommended that you use the default language (such as English) in the XML and use bundle files to include any other supported languages. For more information, see [“Including a manifest of translated text” on page 1112](#)

- Optionally, one or more **variable definitions**, which you can use to have the Workflows task prompt the user for input values. A number of data types are supported for variables, including string, integer, decimal, boolean, time, and date. A declared variable can be referenced by one or more steps in a given workflow. For more information, see [“Defining variables for your workflow” on page 1154](#).
- One or more **step elements** that describe the steps of the workflow. A workflow definition file must include at least one step, and should include all of the steps needed to complete an activity on z/OS (the workflow). For more information, see [“Defining steps for your workflow” on page 1117](#).

The reference tables in [“Workflow XML reference” on page 1173](#) summarize the basic elements of a workflow definition file, including the attribute values, descriptions, any default values, the XML attribute data types, and whether a particular attribute is required.

Creating and viewing the workflow definition file

This topic is intended to give application programmers guidance on how to create a workflow definition file.

To be considered valid, a workflow definition file must follow normal XML syntax conventions and also conform to the rules of the Workflows task schema that is supplied with z/OSMF.

Editing XML files on your workstation

It is recommended that you create and view the workflow XML files on a workstation, rather than on a z/OS system.

When you work with XML files, use a text editor that includes an XML validation function. Validation is the process of comparing your XML files with the Workflows task schema. Doing so ensures that the files use only those tags that are defined in the schema, and ensures that the files conform to the element rules specified in the schema.

To perform the XML validation, you need to transfer the schema file to the XML editor on your workstation. The Workflows task schema resides on the z/OS system according to the z/OSMF product directory path. By default, at this location:

```
/usr/lpp/zosmf/workflow/schemas/workflow_v1.xsd
```

For information about transport protocols, see [“Transferring the workstation files to z/OS” on page 1105](#).

Specifying the processing instruction in the primary XML file

As mentioned in [“Structure of a workflow definition file” on page 1103](#), the primary XML file must begin with a processing instruction in column 1 of line 1. This instruction indicates to the XML processor the version of XML used and the file encoding format. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The following encoding formats are valid:

- UTF-8 (ASCII) or IBM-1047 (EBCDIC) for workflow definition files in UNIX files
- IBM-1047 for workflow definition files in z/OS data sets.

You must include a processing instruction in the primary XML file. However, you do not need to specify a processing instruction in any XML fragment files that you include with the primary XML file.

Some workstation XML editors might not recognize the IBM-1047 processing instruction and therefore do not display the file. Also, it is not possible to display a file in EBCDIC on your workstation. As a workaround, if you want a file to be IBM-1047, you can specify UTF-8 in the processing instruction while you are editing the file on your workstation. Then, after you transfer the file to z/OS in ASCII mode, you can edit the file on z/OS to change the processing instruction to IBM-1047. This action allows the Workflows task to process the file.

Transferring the workstation files to z/OS

It is recommended that you use File Transfer Protocol (FTP) to transfer the XML files to a z/OS system. Doing so helps to ensure that the files are encoded properly for use on z/OS.

For XML files:

- If the processing instruction (the first line in the XML file) indicates that the file uses UTF-8 encoding, transfer the file to z/OS in binary mode, to preserve the file encoding.
- If the processing instruction indicates that the file used IBM-1047 encoding, transfer the file to z/OS in ASCII mode so that the file is converted to EBCDIC.

As supplied by IBM, the Workflows task schema file is encoded in UTF-8. Specify binary mode when you transfer this file to a z/OS system. After you transfer the files to the z/OS system, check the permissions of the transferred files to ensure that they can be opened by the z/OSMF Workflows task. For testing and workflow development purposes, consider setting the file permission of the transferred files to the octal value of 0777 (in "properties").

Saving a workflow definition on z/OS

If you are saving to a data set, ensure that the data set logical record length (LRECL) is large enough to contain the XML file. Otherwise, the save request fails with an error message. For most workflow definition files, an LRECL of 1024 is large enough.

It is possible to save the workflow definition in a different file format, or in a different location. If you attempt to do so, check the workflow definition for any relative references to external file, including references that are represented by substitution variables. For such references, convert each reference to an absolute path (for a UNIX file) or a fully qualified data set name. This change ensures that the external files can be found after the workflow definition is saved in the new format or location.

As a suggested practice, use the Workflow Editor to create, edit, and save workflow definition files. The Workflow Editor can help to ensure that the workflow definition is saved correctly and that file references are converted properly. For more information, see the Workflow Editor online help.

Sample XML files for your reference

To help demonstrate various capabilities of the workflow XML schema, z/OSMF includes a number of sample XML files. It is recommended that you load these samples and observe their behavior in the Workflows task as you read this information.

The samples are supplied with z/OSMF in the /samples subdirectory of the product file system, which is, by default: /usr/lpp/zosmf/samples.

Start with the following samples, which show a basic workflow definition, and demonstrate the use of language bundles and variables:

workflow_sample_basic.xml

Shows the most basic workflow. It contains one step with only the required elements.

workflow_sample_variables.xml

Shows the use of variables that require user input.

More advanced concepts are illustrated in the following samples:

workflow_sample_array_variables.xml

Shows the use of array variables in a workflow.

workflow_sample_array_property.txt

Shows a sample workflow variable input file, for use with workflow_sample_array_variables.xml.

workflow_sample_automation.xml

Shows the use of automated steps in a workflow.

workflow_sample_automation_property.txt

Shows a sample workflow variable input file, for use with workflow_sample_automation.xml.

workflow_sample_calledwfBasic.xml

Shows an example of a called workflow. This workflow is called by specifying its workflow ID.

workflow_sample_calledwfMD5.xml

Shows an example of a called workflow. This workflow is called by specifying its MD5 encrypted value (a 128-bit hash value).

workflow_sample_calledwfVarMapping1.xml

Shows an example of a called workflow. This example shows how variables can be mapped from the calling workflow to the called workflow.

workflow_sample_calledwfVarMapping2.xml

Shows an example of a called workflow. This example shows how variables can be mapped from the called workflow to the calling workflow.

workflow_sample_condition.xml

Shows the use of conditional steps in a workflow.

workflow_sample_feedback.xml

Shows an example of a feedback form that can be used to gather input on a step from the step owner.

workflow_sample_file_template0.xml

Shows the use of a file creation template.

workflow_sample_include_external.xml

Shows the use of a DTD to make references to the external files `workflow_sample_fragment0.xml` and `workflow_sample_fragment1.xml`. This sample also demonstrates other features of steps, and uses some HTML tags within a step description.

workflow_sample_output.xml

Shows an example of writing generated variables to an output file.

workflow_sample_parallel_steps.xml

Shows the use of parallel steps in a workflow.

workflow_sample_program_execution.xml

Shows an example of running an inline executable program (a UNIX shell script) from within a step.

workflow_sample_predefinedVariable.xml

Shows the use of predefined variables in workflow steps.

workflow_sample_rexx_template0.txt

Shows how to invoke a REXX exec from a workflow.

workflow_sample_substeps.xml

Shows the use of substeps and the use of step prerequisites to establish dependency chains.

workflow_sample_translation.xml

Shows a basic workflow that refers to a language bundle file. This workflow is used with `workflow_sample_bundle0.txt`.

workflow_sample_wf2wf.xml

Is a workflow that calls another workflow for processing.

workflow_sample_wizards.xml

Shows the use of instructions and wizards that use input variables.

workflow_sample_wizards_upgrade.xml

Shows the use of the workflow upgrade function. It upgrades the workflow that is created in the sample file `workflow_sample_wizards.xml`.

References to external files

When you refer to an external file from the primary XML file, observe the following considerations:

- If the external file resides in a z/OS data set, specify the fully qualified data set name, which is preceded by a double forward slash (//). Do not enclose the data set name in single quotation marks.

For example:


```
//SYS1.PRODUCTX.TESTFLOW  
//SYS1.PRODUCTX(TESTFLOW)
```

- If the external file resides in a UNIX file, you can specify an absolute (fully qualified) path name, or a relative path name (that is, relative to the primary XML file).

When the workflow definition file is imported into z/OSMF, the Workflows task verifies that each referenced file exists and that the user has READ access to the files. The Workflows task then makes copies of the files, and later refers only to the copies.

Using variable substitution in the workflow definition file path

If you refer to an external file in your workflow definition, you can use variables in the file path name. Doing so allows users of the workflow to customize the file path for their environment. Thus, a file path that uses variables can add flexibility to your workflow definition.

To enable a workflow definition for file path substitution, on the <fileTemplate> element, set the attribute `filePathSubstitution` to `true`. Doing so means that the workflow user is responsible for ensuring that any variables that are used in the file template path must be replaced with valid values.

To supply valid values, the workflow user must edit the workflow input property file and replace the substitution variables with installation-specific values. The user must do this substitution before creating the workflow in the UI. The values that are supplied for the variables in file path are used only during workflow creation time, and cannot be changed during the workflow.

The default value for "filePathSubstitution" is false.

For example, assume that your workflow definition is defined, as follows:

```
<fileTemplate substitution="true" filePathSubstitution="true">  
/u/${instance-filepath}MyTemplate.txt  
</fileTemplate>
```

Here, the workflow user must provide a value for the instance variable "filePath" in the input property file, such as: `filePath=testDir`.

When the user proceeds through the Create Workflow dialog, the Workflows task performs the variable substitution to derive the actual file path for the step: `/u/testDir/MyTemplate.txt`

If your workflow definition is a UNIX file, the <fileTemplate> file path must be a UNIX path. If your workflow definition is a PDS member, the <fileTemplate> file path must be a data set name.

The Workflows task performs validation checking of the file path. A valid file path is one of the following:

- An absolute UNIX path name
- A fully qualified data set name (sequential or PDS) path name (a fully qualified name starting with "///")
- A relative path name, which is relative to the main XML file container. This structure can be a UNIX directory or a PDS. For a PDS, a relative path is the name of a member within the PDS. You cannot specify a relative path when the container is a sequential data set.

Notes:

1. If you do not specify `filePathSubstitution="true"`, the file template path is treated as a UNIX path even if it contains variables. Remember, UNIX systems support most special characters in directory names, such as "\$", "{", "}".
2. On creation of the workflow, the substituted file path is saved as a property to its corresponding workflow step. It cannot be changed during the workflow, regardless of whether the variable is changed later.

Defining entities for a workflow

You can use the document type definition (DTD) of XML to define entities in the workflow definition file. Entities are external files, fragments, or variables that can be referenced within the workflow definition

file. The XML processor replaces the references with the values that are specified in the DTD. You might use an entity to define a value that is subject to change as you develop the workflow, and thus can be changed in one place to affect all references.

You can define entities either inline, in one of the workflow XML files, or as system entities, in which case the XML processor obtains the replacement text from an external file.

Observe the following coding considerations:

- An inline entity can be used like a macro instruction. That is, you can define text in one place that is frequently referenced throughout the workflow definition file.
- A system entity is useful if you want to split the workflow definition file into smaller chunks for manageability, or reuse portions across multiple workflow definition files. For example, for a set of steps that is shared across different workflows. A system entity file can reside in the UNIX file system or in a z/OS data set, and the path name format is as described in [“References to external files” on page 1106](#).

The path name must be expressed as relative (not absolute). A further restriction for DTD entities is that the referenced entity must reside in the same "container" as the main workflow XML file.

- For a UNIX file, the referenced entity must reside in the same directory or a subdirectory of the primary XML file.
- For a PDS, the referenced entity must reside in the same PDS. Here, a *relative path name* is simply the name of the member within the PDS.

Do not use a sequential data set to store an entity file. Also, be aware that a workflow definition that is contained in a sequential data set cannot refer to external entity files.

An entity file in a PDS member must start with the XML processing instruction, as described in [“Creating the workflow definition file” on page 1102](#), with IBM-1047 specified as the encoding format. In fact, any entity file that uses IBM-1047, whether it comes from a data set or a UNIX file, must start with this processing instruction. For UTF-8 files, the instruction is optional.

The following example shows how entities can be defined in the DTD.

```
<!DOCTYPE workflow [<!ENTITY copyright "Copyright IBM Corp., 2013">
    <!ENTITY step1 SYSTEM "step1.xml" >
    <!ENTITY step2 SYSTEM "step2.xml" >
    <!ENTITY step3 SYSTEM "step3.xml" >
]>
```

An entity can be referred to in the document by using the following notation:

```
&copyright;
```

Tip: For another example, see the sample `workflow_sample_include_external.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system. This sample shows the use of a DTD to make references to the external files `workflow_sample_fragment0.xml` and `workflow_sample_fragment1.xml`, which are also supplied in the `/samples` subdirectory. The `workflow_sample_include_external.xml` sample also demonstrates other features of steps, and uses some HTML tags within a step description.

Specifying the workflow root element

Use the workflow root element (`<workflow>`) to specify the XMLSchema-instance namespace, and optionally, a schema location.

For example:

```
<workflow xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="workflow_v1.xsd">
```

The workflow element must end with a closing element: `</workflow>`.

Between the starting and ending element is the body of the workflow definition.

The schema file location specified here is ignored by the Workflows task when it validates an imported workflow definition file. You might find it helpful, however, to use a language-sensitive text editor when creating your workflow definition file. The path you specify indicates the location (relative to the file being edited if expressed as a relative path name) of the schema. Such editors can provide immediate feedback if you violate a rule of the schema, and might provide type-ahead support as you enter elements and attributes.

Though you might develop your workflow primarily on your workstation, and even though your editor is not flagging any errors, you do not know for sure that you have created a valid workflow until after you have imported it into the Workflows task of z/OSMF. Observe the following coding considerations:

- Differences exist among the various implementations of XML schema validators. Thus, a workflow definition file that validates in your text editor might not validate when it is imported into the Workflows task.
- The Workflows task performs additional validation that is not enforced by the schema, for example, validating that any message or template files that are referenced in the workflow actually exist on the z/OS system.

Specifying the workflow metadata

Use the workflow information element (<workflowInfo>) to specify metadata for the workflow, such as the workflow identifier, description, version, and vendor, and possibly other details. The Workflows task displays the metadata to users in the Workflows table and the **Properties for workflow** page. Users can use the metadata values for filtering or configuring the workflows view.

The workflow information element is required.

You can specify the following sub-elements for the workflow information element:

parallelSteps

For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, set this property to `true`. Otherwise, if this property is omitted or set to `false`, automated steps are run one-by-one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

maintainFragments

Fragments are valid XML files that can contain any combination of elements that follow the workflow schema. Fragments are defined by external entity references at the beginning of the definition file. By default, when changes are saved all fragment content is merged into a single file single XML file that no longer contains references to the fragment files. When this property is set to `true`, fragment references in the workflow definitions are instead preserved, and any updates are applied to all fragment files when changes are saved.

The following conditions must be followed for this service to be supported:

- XML definition files must be pre-parsed or follow parsing standards. Each element must be logically separated by new lines.
- Comments nested in XML elements are not maintained.
- Fragment references through variable entities or parameter entities are not supported.
- Nested fragments are not supported.
- The workflow cannot be a data set.
- The same fragment cannot be used twice in the same workflow definition.
- The **Save as** action in Workflow Editor cannot be used while this function is enabled.

workflowID

Short, arbitrary value that identifies the workflow. This element is required.

This element can include the optional attributes *scope* and *isCallable*. For more information, see [“Setting the workflow scope” on page 1111](#) and [“Callable workflows” on page 1111](#).

workflowDefaultName

Default name for the workflow. The default name is shown in the **Workflow name** field of the Workflows task when a user creates the workflow. The user can accept this name, or specify a different name.

The workflow name:

- Must be unique in the Workflows task.
- Can contain up to 100 characters. Leading and trailing white space is ignored.
- Must not contain the characters for ampersand ('&'), forward slash ('/'), logical or ('|'), greater than ('>'), or less than ('<').

The workflow name is not case-sensitive; for example: MyWorkflow and MYWORKFLOW are the same workflow.

This element is optional. If you omit it, the Workflows task assigns a name to the workflow, based on the following convention:

```
<workflow-description> - Workflow_<number>
```

Where:

- *workflow-description* is the description from the workflow definition file.
- *number* is the first available number, beginning at 0. If you later delete this workflow, its number can be reused by the Workflows task.

workflowDescription

Short description of the workflow. This element is required.

When a workflow is imported into z/OSMF, the Workflows task uses the workflow description (specified in the <workflowDescription> tag) to create a default workflow name, which the user can change.

workflowVersion

Version of the workflow definition. This element is required.

The Workflows task caches only the latest version of any imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you change any portion of the workflow definition. This includes changes to the primary XML file or any subfiles or referenced files. For this reason, when you author a workflow definition, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.

vendor

Name of the workflow provider. This element is required.

jobsOutputDirectory

Indicates a location to be used for automatically storing saved job output files from the workflow. Use this option if you want to retain job output files, perhaps as a record of the work that is done by the workflow. Specify a valid UNIX file path and directory on the user's system, beginning with a single forward slash ('/'). For example: /u/IBMUUSER/jobFiles. In the Workflows task user interface (UI), the workflow owner can overwrite this value with a different location, as needed.

If this option is enabled, a value must be specified. The value must be an existing directory on the user's system. Otherwise, the Workflows task cannot create the workflow; an error message is displayed in the Workflows task UI.

The job output files are saved in the IBM-1047 encoding format. This format is viewable in a z/OS Console by using the **cat** command. As an alternative, the user can use an FTP tool to view or download the files with the transfer type ASCII.

Observe the following authorization requirements:

- Workflow owner user ID requires write access to the directory.

- For the steps that create job output, the step owner user IDs require write access to the directory. Otherwise, the steps cannot be performed.

If the Job Output Location option is not enabled, the workflow does not save its job output files.

Table 586 on page 1175 describes the elements that make up the workflow metadata.

Figure 489 on page 1111 shows an example of the metadata XML tags for a workflow.

```
<workflowInfo>
  <parallelSteps>true</parallelSteps>
  <workflowID scope="system" isCallable="system">SampleWorkflow</workflowID>
  <workflowDefaultName>WORKFLOW#001</workflowDefaultName>
  <workflowDescription>A simple workflow</workflowDescription>
  <workflowVersion>1.0</workflowVersion>
  <vendor>XYZ, Inc.</vendor>
  <General/> <!-- This element is empty, and completely optional. -->
  <jobsOutputDirectory>/u/IBMUUSER/jobFiles</jobsOutputDirectory>
</workflowInfo>
```

Figure 489. General metadata for a workflow

Setting the workflow scope

It is possible to have multiple instances of a workflow run at the same time, on the same system or sysplex. However, in some cases you might need to restrict a workflow to one instance only, which can be useful for coordinating actions across a system or sysplex. In z/OSMF workflows, this concept is referred to as the *singleton scope* for the workflow.

To set the singleton scope for a workflow, specify the optional attribute *scope* on the sub-element (`<workflowID>`). Specify the scope attribute with one of the following values:

system

A maximum of one instance of this workflow can exist on any one system in the sysplex.

sysplex

A maximum of one instance of this workflow can exist in the sysplex.

none

No limit exists for the number of instances of this workflow. For a callable workflow, this setting means that a new instance is always created on the calling system.

Workflow scope has ramifications for callable workflows. You might, for example, determine that a new instance of the workflow should always be created in response to a calling workflow. Or, you might prefer to have an existing instance of a workflow be used to handle the request. For considerations, see [“Coordinating workflow-to-workflow actions” on page 1138](#).

Callable workflows

In z/OSMF, a workflow can invoke another workflow for processing. A workflow that invokes another workflow is the *calling workflow*. A workflow that is invoked by another workflow is the *called workflow*.

To indicate whether a workflow is eligible to be called by another workflow, specify the optional attribute *isCallable* on the workflow sub-element (`<workflowID>`). On the *isCallable* attribute, also indicate the callable *range* for the workflow, that is, whether the workflow can be called only within the same system, or whether it can be called from any system in the same sysplex. Specify the *isCallable* attribute with one of the following values:

system

This workflow can be called only by another workflow that is running on the same system.

sysplex

This workflow can be called by any workflow that is running in the same sysplex.

If you omit the `isCallable` attribute, the workflow cannot be called by another workflow.

More information about callable workflows is provided in [“Calling steps” on page 1137](#).

Workflow category

In z/OSMF, a workflow category is a classification of the activities that are to be performed in a particular workflow. Workflows that are used to configure system software such as a product or component are *Configuration* workflows. Workflows that are used to provision system software, such as Db2 or IMS, are *Provisioning* workflows. All other workflows are *General* workflows.

The category is specified by the workflow author in the workflow definition file. The workflow category is optional. If no category is specified for the workflow, the workflow belongs to the General category. [Figure 489 on page 1111](#) shows an example of how the workflow category is specified: `<General/>`.

For a Configuration workflow or a Provisioning workflow, the workflow metadata specifies product-specific details, such as the product name, product version, and product ID. The metadata for a Provisioning workflow also specifies the type of software to be provisioned (software type). The Workflows task presents this information to users in the **Category Specific** tab on the **Properties for workflow** page.

The following example shows the metadata for a Configuration workflow:

```
<workflowInfo>
  <workflowID>Productx-01</workflowID>
  <workflowDefaultName>WORKFLOW#001</workflowDefaultName>
  <workflowDescription>Initial configuration of Product X</workflowDescription>
  <workflowVersion>1.0</workflowVersion>
  <vendor>IBM</vendor>
  <Configuration>
    <productID>abc123</productID>
    <productName>Product X</productName>
    <productVersion>V1R99</productVersion>
  </Configuration>
  <jobsOutputDirectory>/u/IBMUUSER/jobFiles</jobsOutputDirectory>
</workflowInfo>
```

In this example, the metadata provides product-specific fields in the `<workflowInfo>` element, as follows:

productID

Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).

productName

Name of the product or component that is being configured through the workflow.

productVersion

Version and release of the product or component that is configured through the workflow.

Including a manifest of translated text

Use the translated text files element (`<translatedTextFiles>`) to define the *message manifest* for a workflow. The message manifest contains one or more bundle definitions, each of which contains one or more language file definitions. A *bundle* is a logical grouping for a set of text elements used in the workflow.

To refer to these files, other text elements in the workflow definition can specify a bundle name and an identifier (or key) within the language files that contain the replacement text for that language. The Workflows task maps the bundle name and language into the file defined for that language and displays the text in the language that is in effect for the browser, if a language file is provided.

The Workflows task uses only the browser language to locate the translation file; the country is ignored. For example, en-US and en-UK are two versions of English: American, and British, respectively. The Workflows task uses only the value "en" to locate the language file.

The following example shows a message manifest:

```
<translatedTextFiles>
  <bundle name="StepMessages">
    <language name="en" path="steps/english.txt"/>
    <language name="fr" path="steps/french.txt"/>
  </bundle>
  <bundle name="VariableLabels">
    <language name="en" path="vars/english.txt"/>
    <language name="fr" path="vars/french.txt"/>
  </bundle>
</translatedTextFiles>
```

For an example of referencing a bundle in a translatable element, see the sample file **workflow_sample_translation.xml**, which is supplied with z/OSMF in the /samples subdirectory of the product file system.

Table 590 on page 1185 describes the elements that make up the message manifest.

Enabling a workflow definition file for future upgrades

Over the course of time, you might want to provide users with a revised version of your workflow, presumably with enhanced functions. This topic describes the tags that you can use to make your workflow definition upgradeable.

In z/OSMF, to *upgrade* a workflow means to create a new instance of an existing workflow, based on a new definition file. As the workflow author, you can design your workflow definition to be upgradeable to future versions of the definition.

During an upgrade operation, the Workflows task user upgrades an existing workflow to a new level of the workflow by using the action **Create New Based on Existing**, which is provided in the Workflows table. By upgrading the workflow, the user creates a new instance of the workflow, while retaining data from the existing workflow.

Creating an upgradeable workflow definition

Using a set of tags in the Workflows schema, you can add an upgrade capability to your workflow definition. The tags specify which previous versions of the workflow are supported for upgrades. Also, the rules for data mapping, which variables, steps and attributes can be copied forward, and so.

To provide upgrade option in a workflow definition, you must provide a workflow definition with same workflowID as the current (old) version of the workflow definition.

To add an upgrade capability to your workflow definition, you define an upgrade element (<preserveOptions>) with the attributes described in this topic.

Start by defining the <preserveOptions> element after all of the defined steps in the new workflow definition. On the <preserveOptions> element, you can specify the following elements and attributes:

- **Version.** This element specifies information about the workflow to be upgraded. It contains the following attributes:
 - **Value.** Version of the workflow definition that is supported for upgrading. You must specify at least one prior version of the workflow definition.
 - **Type.** Specify either patch or release, as follows:
 - **Patch** indicates that the upgrade to is intended to fix a defect in the prior version of the workflow. Based on this idea, the new workflow replaces, rather than coexists, with the prior version. Thus, the prior version is canceled when the upgrade is performed.

- **Release** indicates that the upgrade creates a new release of the workflow. The prior version is retained, based on the assumption that some installations with multiple releases might choose to use more than one version of the workflow.
- Variable set (<variableSet>). This element specifies which variable values are to be copied from the existing workflow. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default) causes the option **Copy attribute values based on upgrade definition** in the Workflows task wizard **Create New Based on Existing** to be selected by default. If so, all of the variable values are copied to the upgraded workflow.
- Step set (<stepSet>). This element specifies the step values to be copied from the existing workflow. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default) causes the option **Copy step attributes based on upgrade definition** in the Workflows task wizard **Create New Based on Existing** to be selected by default. If so, all of the current workflow step attributes are copied to the upgraded workflow. The attributes include the step assignees, step owners, step states, notes, submitted job status, and the record of how variable conflicts, if any, were resolved.
- Workflow history (<workflowHistory>). This element specifies whether the workflow history is copied from the existing workflow to the new instance. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default), causes the option **Copy workflow history** in the Workflows task wizard **Create New Based on Existing** to be selected by default. Otherwise, the option is not selected.
- Workflow notes (<workflowNotes>). This element specifies whether workflow notes are copied from the existing workflow to the new instance. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default), causes the option **Copy workflow notes** in the Workflows task wizard **Create New Based on Existing** to be selected by default. Otherwise, the option is not selected.
- Include (<include>). Use this element to specify which steps or variables in the prior workflow definition file are copied to the new workflow. You can specify this element multiple times. Specify either a regular expression or a variable name.
- Exclude (<exclude>). Use this element to specify which steps or variables are excluded from the set that is generated by the <include> element.
- Upgrade notes (<upgradeNotes>). Use this element to provide the user with information about the upgraded workflow definition. For example, you might use this element to provide the user with details about using the workflow.

For descriptions and data types of these elements and attributes, see [“Workflow upgrade elements summary”](#) on page 1182.

The <include> element in the variable set specifies which variables are to be copied, based on either a regular expression (regExp) or the variable name. The following example specifies that all of the existing steps that match the specified regular expression are to be copied to the same steps in the new workflow definition:

```
<include regexp="\w★">
```

To copy the value from the variable that is named *setting3* in an existing workflow to the same variable in the new workflow definition, you might code the following:

```
<include name="setting3"/>
```

To copy the value from the variable that is named *setting1* in an existing workflow to the variable named *setting2* in the new workflow definition, you might code the following:

```
<include name="setting1" mapTo="setting2"/>
```

If you specify the <include> element multiple times in a sequence, and a variable is included for multiple times after calculation by name and regular expression, the last <include> specification is the one that is used for the copy operation.

Collecting user feedback

It is possible to collect feedback from the users of a workflow. A workflow author can optionally include a feedback form on one or more steps with customized questions for the step owner to answer at the conclusion of a step. Such feedback can be useful for determining the effectiveness of a workflow design, or collecting user requirements for future enhancements to a workflow. Inclusion of a feedback form is optional; answering the questions in a feedback form can be optional or required, as determined by the workflow author.

The workflows XML schema includes elements to help workflow authors create questions for users. The Workflows task includes functions to allow the workflow owner to prompt users for feedback, and collect the responses into a consolidated document. When all of the required feedback is provided, the workflow owner can send the feedback to the workflow author for evaluation.

How feedback is collected

Collecting feedback from the users of a workflow involves the participation of the following roles:

Workflow author

Workflow author defines feedback questions in the workflow definition file, and designates the questions as optional or required.

Workflow owner

For a workflow that includes feedback questions, the workflow owner is responsible for collecting the feedback. From the Workflows table, the workflow owner can select Feedback to launch actions that are related to feedback. The workflow owner can display pages to see which steps require feedback, which steps have incomplete feedback, and options for notifying the step owners who need to complete feedback.

Step owner

For a step that includes a feedback form, the step owner is responsible for providing feedback by answering questions about the step. To answer feedback for a step, the step owner selects a step and selects the table action Feedback. Only the step owner can display the feedback page for a step. The Feedback action is disabled for steps that do not contain feedback questions.

When all of the required feedback is provided by step owners, the workflow owner can save the accumulated feedback into a feedback file. On the Generate Feedback Summary page, the workflow owner can create a report of the feedback, which can be sent to the workflow vendor for evaluation.

Schema elements

As the workflow author, you define feedback questions and answers in the workflow definition file.

Example

[Figure 490 on page 1116](#) shows how to define each of the question types in a workflow definition file.

In the example:

- `itemOne` defines a multiple choice question
- `itemTwo` defines an either or choice
- `itemThree` defines a question that accepts a write-in response.

```

<!-- ===== feedback definitions ===== -->
<feedbackItem name="itemOne">
  <question>How difficult was this step?</question>
  <!--The user must select one answer from the available choices-->
  <answers>
    <singleSelect hasOtherAnswer="true">
      <label value="difficult">Very difficult</label>
      <label value="moderate">Somewhat difficult</label>
      <label value="moderate">Neutral</label>
      <label value="moderate">Somewhat easy</label>
      <label value="easy">Very easy</label>
    </singleSelect>
  </answers>
</feedbackItem>

<feedbackItem name="itemTwo">
  <question>What did you like about this step?</question>
  <!--The user can select more than one of the available choices-->
  <answers>
    <multipleSelect hasOtherAnswer="true">
      <label value="simple">Ease of use</label>
      <label value="info">Instructions were helpful</label>
      <label value="useful">Performed a useful function</label>
      <label value="quick">Ran quickly</label>
    </multipleSelect>
  </answers>
</feedbackItem>

<feedbackItem name="itemThree">
  <question>How would you describe your experience with this step?</question>
  <!--The user supplies a text response (up to 500 characters) -->
  <answers>
    <text/>
  </answers>
</feedbackItem>

```

Figure 490. You can define various questions for step owners to answer.

The questions that are defined in Figure 490 on page 1116 can be referenced by the steps in your workflow. Figure 491 on page 1116 shows how the questions defined earlier can be included in the step definitions. Notice the attribute `required` is included for questions that require an answer from the step owner.

```

<!-- ===== step with feedback questions ===== -->
<step name="StepOne" >
  <title>A step with feedback</title>
  <description>A step with feedback.</description>
  <!--On the feedback tag, the attribute "name" identifies the
    feedback question. The attribute "required" indicates whether
    a user response is required or optional. -->
  <feedback name="itemOne" required="true"/>
  <feedback name="itemTwo" required="true"/>
  <feedback name="itemThree" required="true"/>
  <instructions>This step has three questions for you to answer.
    All of the questions require a response.</instructions>
  <weight>1</weight>
</step>
:
<!-- ===== Another step with feedback questions ===== -->
<step name="StepTwo" >
  <title>Another step with feedback</title>
  <description>Another step with feedback.</description>
  <!--On the feedback tag, the attribute "name" identifies the
    feedback question. To indicate that a user response is required,
    the attribute "required" is included and set to true. To indicate
    that a response is optional, you can set required to false or
    omit this attribute. -->
  <feedback name="itemOne" required="true"/>
  <feedback name="itemThree"/>
  <instructions>This step has two feedback items. itemOne is required,
    and itemThree is optional.</instructions>
  <weight>1</weight>
</step>

```

Figure 491. How feedback questions can be included in the steps in your workflow.

Defining steps for your workflow

A workflow is composed of one or more units of work called *steps*. A workflow definition file must contain at least one step; each step can contain substeps. In the Workflows task, a wizard guides the user through a step, which can be either manual or automated. For a manual step, the wizard only displays the instructions that are required for the user to perform the step. For an automated step, the wizard uses a template to create a file or run a REXX exec, UNIX shell script, or JCL job. Optionally, the wizard can perform substitution by referencing variable values.

Every step has a name and contains a title and description. Steps can contain substeps, which can also contain substeps, up to five levels of nesting. For example, Step 1 can contain Step 1.1, which can contain Step 1.1.1, which can contain Step 1.1.1.1, which can contain Step 1.1.1.1.1. The total aggregation of steps and their substeps across the entire workflow cannot exceed 500.

The step name must be unique across the entire workflow. The step name is not displayed in the Workflows task, but it is used within the workflow to reference prerequisite steps. The title should be brief. It is displayed in the step table when the workflow is opened. Step titles are indented for substeps. The step description can be more detailed. It is displayed in the "General" tab under step properties to provide a bit more context, if necessary.

A step can be designated as optional. This designation has effects on how its weight is used in the calculation of the percentage-complete for a workflow. Weights are described in [“Parent steps and leaf steps” on page 1117](#).

Any step can optionally contain a set of references to prerequisite steps (by step name) which must be completed before this step can be performed. Each prerequisite is identified in its own `<prereqStep>` element. The Workflows task displays the prerequisite chain for a step in the "Details" tab for the step properties. The Workflows task also keeps track of dependencies with regard to the states of the steps.

Any prerequisite step must have been defined previously in the workflow (though not an ancestor of the referencing step). If the reference is to a step with substeps, the substeps are treated as prerequisites (they need not be explicitly specified). If only certain substeps under a parent step are dependencies, they can be listed explicitly.

A step can contain multiple prerequisite steps, but these need not be cumulative. That is, if Step3 depends on Step2, and Step2 depends on Step1, then Step2 would identify Step1 as a dependency, and Step3 would identify Step2 as a dependency. If Step3 identified both Step2 and Step1, no harm would occur, but it makes the workflow more complex than necessary, and possibly more difficult to maintain over time.

A step can be defined as *conditional*. Such a step is available to be performed based on whether a logical condition is satisfied on the z/OS system. For example, a conditional step might become eligible to be performed if a job run by another step ends with a particular return code. A conditional step, which depends on a logical condition, is different than a dependent step, which depends on a particular step being completed. For more information, see [“Making a step conditional” on page 1147](#).

What is described here applies to any step. For the distinctions between a parent step and a leaf step, see [“Parent steps and leaf steps” on page 1117](#).

[Table 591 on page 1187](#) describes the elements that make up a step.

Parent steps and leaf steps

A *parent step* contains a set of nested step elements (at least one). A step with no substeps is called a *leaf step*.

A leaf step actually performs, or tells the user how to perform, the actions required to complete the step. Leaf steps contain, at a minimum, instructions and a weight. Optionally, a leaf step can also contain a skills category, a template (for file creation or job submission), and references to variables which can be substituted into the instructions or template, or both.

The skills element (<skills>) specifies a suggested skills category for performing the step, such as "Security administration" or "Network administration." The Workflows task displays this value in the step table for a workflow. This value is free-form; specify it at your discretion.

The weight element (<weight>) specifies the relative difficulty of the step (a positive integer value from 1 to 1000). The Workflows task uses this value in the calculation of the percentage-complete value that is displayed as the workflow is performed. The scale is arbitrary; specify it at your discretion. Specify a lesser value for a step in which the user performs a simple action, such as cutting and pasting some a command text that you provide. For a more complicated task, such as deploying a digital certificate on z/OS, specify a greater value.

The instructions element (<instructions>) defines the content of the *Review Instructions* tab of the Step Perform wizard in the Workflows task. In this element, you provide the detailed instructions on what the user must do to perform the step to completion. The instructions can contain certain HTML tags for formatting, and hyperlinks to refer to additional information. The instructions can also use variable substitution. The <instructions> element is required, and must contain some content. Otherwise the step cannot be performed (the Workflows task Perform tab is omitted for the step).

In the Workflows task, the *Review Instructions* tab is displayed after variable-prompting. Thus, instructions can contain substituted values. However, instructions cannot be used to guide the user through the variable gathering stage. The variable attributes (title, abstract, and description) must be sufficient to guide the user, though you can also provide some guidance or context in the step description.

Review Instructions is the final tab that the user sees when performing a manual step. The user is expected to follow the instructions and then press **Finish** to complete the step.

Usually, a step must be marked complete before the workflow can continue. Depending on your design, however, you might want to allow the user to mark a step as *Failed* manually. This option might be useful if a step cannot be performed manually (outside of the workflow). To enable this option, include the can-mark-as-failed element (<canMarkAsFailed>) for the step and set it to true. Doing so causes the *Review Instructions* tab to prompt the user to confirm whether the step could be completed manually. If appropriate, the user can mark the step as *Failed* and continue with the workflow. By default, this option is disabled (the can-mark-as-failed element is set to false).

Leaf steps can contain optional references to variables that were defined earlier in the workflow definition. Based on your requirements, you might need to allow the step owner to modify the value of a referenced variable. To do so, identify the variable reference with the variable value element (<variableValue>). If so, the Workflows task displays a wizard to guide the user through entering values for the variable when performing the step, as described in [“Defining variables for your workflow” on page 1154](#). If a step only references a variable for read and not for modification, you do not need to specify the <variableValue> element.

Note: Array variables, which are used to map multiple values, are not shown in the Workflows task wizard.

You can code symbols using the Velocity syntax in both the instructions and the template that can be replaced by the value of the variable as entered by the user. More information is provided in [“Using Velocity templates for variable substitution and other functions” on page 1155](#).

You can determine how and when your workflow should prompt the user for variable values. For instance, you could have a step that is used only to prompt for the variables, which are then referenced by subsequent steps. Or, you can prompt for the variable directly within the step that uses it.

When coding the variable reference within a step, you can specify whether the variable is required. If a variable is required, and the user does not enter a value for it, the Workflows task prevents the user from proceeding until a value is specified. If a variable is optional, and the user does not enter a value for it, the Workflows task allows the user to advance to the edit screen, and the Velocity symbol in the instructions or template is displayed as-is without substitution. You can, however, use conditional Velocity statements to generate different text based on whether a variable has a value. If the symbol is left unresolved, a user might not understand these references, depending on how you name the variable. An unresolved variable in a template will likely result in an incorrect file or executable. However, the user can optionally edit the files before saving them or running them.

The variable reference also specifies whether to allow the user to change the value if the variable has one already. As a workflow author, you might know that the variable has already been used for an important purpose, such as allocating a file, and that, if the user changes the value, the overall procedure defined in the workflow would be corrupted. Here, you should not allow the user the opportunity to change the value. Instead, the Workflows task displays the value in read-only mode. The user can still override the value, after responding to a confirmation prompt that includes a list of any other steps that reference the same variable. For any variable that is referenced by additional steps, the Workflows task displays a list of those steps, along with the variable description, when the user presses the information icon. Even if you make a variable read-only when it already has a value, the user can override the value by checking a box in the description to make the value editable, thus accepting the consequences of doing so.

Template steps

A step that runs a program, such as a JCL job, a REXX exec, or a UNIX shell script, is referred to as a *template step*. The program that is run by the step is referred to as a *template*. This topic describes how to write a template step so that you can run programs and batch jobs in your workflow.

Program design considerations

For a template step, you must determine:

- Whether to include the executable code inline (within the step XML tagging), or in a separate, external file that is referenced from the step.
- How the executable code (the template) is processed — in real time or submitted as a batch job. You can:
 - Run a program in real time for immediate results. The program can be a REXX exec or a UNIX shell script.
 - Submit a job for batch processing. The batch job contains JCL and might also imbed an executable program that runs under batch, such as a REXX exec or UNIX shell script.

For more information, see [“Running a program in real time” on page 1125](#) and [“Submitting a JCL job for batch processing” on page 1128](#).

- Whether the program uses variables (substitution values). You can use variables to prompt for input from the Workflows task user. In the Workflows task user interface (UI), the variable value is presented to the user, who can override it. For more information about variables, see [“Using variables in a template step” on page 1123](#).
- Whether the program contents can be modified by the end user, and if so, where the modified template is saved. For more information, see [“Saving the contents of the template” on page 1123](#).
- Whether the program creates a properties file, which contains variables for subsequent steps to reference. For more information, see [“Creating a properties file” on page 1124](#).

How the user interacts with a template step

When a step includes a template element, the Workflows task user interface (UI) enables the **Next** button on the wizard instructions page. When the user presses **Next**, the wizard guides the user through the activity, such as running a program or job, or creating and saving a properties file. The behavior of the wizard is further controlled through the elements that you define within the template element.

How automation processes a template step

For an automated workflow, the Workflows task performs the step under the user's identity. As a result, the user can view the results of the step processing, such as a job status or a properties file.

The user's identity is either the step owner user ID or a runAsUser user ID, if the element runAsUser (<runAsUser>) is included on the step element. When a runAsUser is not specified for a step, the step is performed under the step owner user ID. For more information about the runAsUser element, see [“runAsUser identity for a step” on page 1149](#).

Elements of a template step

In a workflow definition, a template step is defined with the element `<template>` and its related elements and attributes in the Workflows XML schema. The elements are used to indicate where to find the program that is run, how the program is processed (submitted in batch or executed in real time), and other details. [Table 575 on page 1120](#) provides a summary of the template step elements.

How the program is processed — immediately or in batch — determines which `<template>` elements are applicable for use with the `<template>` element. Indicate your choice of program processing by using the element `<submitAs>` with the appropriate setting. In [Table 575 on page 1120](#), the applicable elements for each processing type are indicated in the columns **Used in an immediate execution step** and **Used in a batch execution step**.

Table 575. Summary of template step elements			
Element Name	Description	Used in an immediate execution step	Used in a batch execution step
template	Identifies the step as a template step, which is a step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script.	Yes	Yes
inlineTemplate	<p>Indicates that the program is included inline, that is, within the step XML structure. This element is mutually exclusive with the <code><fileTemplate></code> element.</p> <p>In the following example, a REXX program is provided inline within the <code><inlineTemplate></code> element.</p> <pre><template> <inlineTemplate substitution="true">/* rexx */ parse arg arg1 SAY "this is a sample to submit TSO REXX script to run immediately" SAY "the first parameter is :" arg1 SAY \${instance-st_user} SAY "outvar:st_group =" SYS123 SAY "outvar:st_user =" USERS SAY "This execution will meets timeout." </inlineTemplate> </template></pre>	Yes	Yes

Table 575. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
fileTemplate	<p>Indicates that the program is contained in an external file, and specifies the location of the file. This element is mutually exclusive with the <code><inlineTemplate></code> element.</p> <p>Specify the location as either the full path name of the file, beginning with the forward slash (/) and including the file name, or a relative path. If the file resides in the same directory as the workflow definition, specify the file name only.</p> <p>In the following example, the program to be executed (workflow_sample_rexx_template0.txt) resides in the same UNIX directory as the workflow definition file:</p> <pre><template> <fileTemplate substitution="true"> workflow_sample_rexx_template0.txt </fileTemplate> <submitAs maxRc="0">TS0-REXX-JCL</submitAs> </template></pre>	Yes	Yes
submitAs	Indicates the type of program processing to be performed, as described in “Running a program in real time” on page 1125 and “Submitting a JCL job for batch processing” on page 1128.	Yes	Yes
maxLrecl	<p>Specifies the maximum record length, in bytes, for the input data for the job. This value is used when the step is performed automatically (autoEnable=true).</p> <p>If the step is performed manually, the user can optionally specify the maximum record length on the Edit JCL page in the Workflows task UI.</p> <p>For more information, see “Maximum record length” on page 1130.</p>	No	Yes
saveAsUnixFile	<p>Specifies the path name for saving the executable code as a UNIX file after the user edits the template in the Workflows task. Must be specified as a full path name of the file, beginning with the forward slash (/) and including the file name.</p> <p>The presence of this element results in the save as UNIX file option being presented to the user, primed with the element value, if specified.</p> <p>For more information, see “Saving the contents of the template” on page 1123.</p>	Yes	Yes

Table 575. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
saveAsDataset	<p>Specifies the data set name for saving the executable code after the user edits the template in the Workflows task. Must be specified as a fully qualified data set name, without quotations.</p> <p>The presence of this element results in the save as data set option being presented to the user, primed with this value, if specified.</p> <p>For more information, see “Saving the contents of the template” on page 1123.</p>	Yes	Yes
successPattern	<p>Specifies a regular expression that is returned for a successful program execution. This element is required. You must specify one (and only one) regular expression for a successful program completion. For more information, see “Completion messages” on page 1126.</p>	Yes	No
failedPattern	<p>Optionally specifies a regular expression that is returned for program execution failures. You can omit this element or specify up to 100 different specifications for <failedPattern>. For more information, see “Completion messages” on page 1126.</p>	Yes	No
outputVariablesPrefix	<p>Specifies a prefix for variables that are created by the step. For more information, see “Writing variables to a properties file” on page 1125.</p>	Yes	Yes
scriptParameters	<p>Contains the input parameters that can be set by the step owner. This element is optional. For more information, see “Input parameters” on page 1125.</p>	Yes	No
procName	<p>Specifies the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC. For more information, see “TSO/E address space for program execution” on page 1126.</p>	Yes	No
regionSize	<p>Specifies the region size for the TSO/E address space. If no value is specified for the step, the default is 50000. For more information, see “TSO/E address space for program execution” on page 1126.</p>	Yes	No
timeout	<p>Specifies the maximum amount of time that the program can run before it is ended by a timeout condition. For more information, see “TSO/E address space for program execution” on page 1126.</p>	Yes	No

Table 575. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
output	Specifies the default name and location of the properties file that is created by the step. The properties file can contain variables and values that are used by subsequent steps. The Workflows task UI allows the user to modify the file name and location, as needed. For more information, see “Creating a properties file” on page 1124 .	Yes	Yes

Notes:

1. Avoid creating an inline template with characters that interfere with XML. If this problem occurs, use a file template instead. Also, when you use Velocity comparison operators in the instructions, do not use the less than (<) or greater than (>) characters, as they interfere with XML. Instead, use the alternative notation: lt, le, gt, and ge.
2. No white space manipulation occurs for an inline template. Use care when you type the template into the XML, based on the context of the template. To write JCL, for example, start the first line of the template immediately after the beginning <inlineTemplate> element. Subsequent lines must start on column 1, rather than being indented for readability.

Using variables in a template step

The "substitution=" attribute on the template element indicates whether the template contains variables. The default is false. If you specify "substitution=true," you must also specify at least one variable value element (<variableValue>) in the step. Otherwise, the workflow fails with a validation error when the user attempts to import the workflow definition into the Workflows task.

Observe the following considerations:

- If you include more variables than you reference, no errors result.
- If you refer to variables that are not included, but you do include at least one variable, no validation errors result. However, the Workflows task cannot resolve the variables during the substitution process.

Saving the contents of the template

To specify the location of an external file for saving the template, include the <saveAsDataset> element or the <saveAsUnixFile> element for the step. The location can be a sequential data set, partitioned data set (PDS), or z/OS UNIX path and file name. The Workflows task UI presents the user with the option to save the template in this location.

By including the <saveAsDataset> element or the <saveAsUnixFile> element, you control whether the template is saved to a data set or a z/OS UNIX file, respectively. To allow the user to save to either data format, include both elements in the step definition.

The template contents are always saved in EBCDIC encoding (code page IBM-1047).

Providing neither element has the same result as defining both elements without a value. Consider it a shorthand way of indicating that you can tolerate the file in either a data set or a UNIX file, and you have no suggestion as to what to name the output. For example, if you do not care about which data set the user selects for saving the program, you can provide an empty element like this: <saveAsDataset/>.

You can set these elements to a default location. If so, the location is initialized in the widget that is presented to the user, who can override it. If you omit a default value, the user must specify a valid location.

Table 576 on page 1124 provides more details about these elements.

Table 576. Elements for saving a generated job	
Element Name	Description
saveAsDataset	<p>Specifies the data set name for saving the executable code after the user edits the template in the Workflows task. Must be specified as a fully qualified data set name, without quotations.</p> <p>The presence of this element results in the save as data set option being presented to the user, primed with this value, if specified.</p>
saveAsUnixFile	<p>Specifies the path name for saving the executable code as a UNIX file after the user edits the template in the Workflows task. Must be specified as a full path name of the file, beginning with the forward slash (/) and including the file name.</p> <p>The presence of this element results in the save as UNIX file option being presented to the user, primed with the element value, if specified.</p>

The <saveAsDataset> and <saveAsUnixFile> elements accept variable substitution. Suppose, for example, that you want to provide a default data set name for saving a job, but also allow the user to specify the data set high-level qualifier. Here, you could specify the <saveAsDataset/> element like this:

```
<saveAsDataSet substitution="true">${instance-hlq}.MYPROD(CONFIG)</saveAsDataset>
```

If you plan to save to a data set, be aware that the data set must already exist (be allocated) when the workflow runs. Otherwise, an error results. The Workflows task does not allocate the data set that you specify on the <saveAsDataset> element. Similarly, the path and file that you specify on <saveAsUnixFile> element must exist when the workflow runs. Otherwise, an error results.

Creating a properties file

You can design a step to run a job to create variables, and save them to a *properties file*. When a step creates a properties file, the Workflows task reads in the contents of the file and saves its values for use by other steps in the same workflow.

To specify the default name and location of the properties file that is produced by the step, include the element <output> on the step element. The properties file can be a data set, a UNIX file, or a JES spool file, as follows:

Data set

Specify a fully qualified data set name, without quotations.

Omit the sysoutDD= attribute from the output element.

UNIX file

Specify a full path name of the file, beginning with the forward slash (/) and including the file name.

Omit the sysoutDD= attribute from the output element.

JES spool file

Include the sysoutDD= attribute on the output element and set this attribute to true.

Specify the spool file by using the following syntax:

```
[step.]ddname
```

Where:

- *step* is the name of the job step that creates the spool file. This value is optional. If no job step name is specified, z/OSMF attempts to find a matching job step by checking the results of the executed job. z/OSMF reads from the most recently created DD name to the earliest to find a match. To avoid ambiguity, it is recommended that you always specify the step name.
- *ddname* is the DD name for the step. This value is required.

When a step includes an output element, the Workflows task UI enables the **Edit Output File Path** option in the step perform wizard. The Workflows user can modify the file name and location, or choose a different file, as needed. If the `sysoutDD=` attribute is set to true, the Workflows task UI enables the **Edit Output DD Name** page in the step perform wizard. The Workflows user can modify the DD name, as needed.

When a properties file is processed, the Workflows task scans the file to determine whether the file contains any workflow input variables (properties that are written as name-value pairs). If so, the Workflows task attempts to add the variables to the variables pool. The input variables then become available for use by subsequent steps in the workflow.

Notes:

1. The identity of the user who performs the step is used to read the resulting properties file. If the step is performed under a `runAsUser` user ID, the `runAsUser` ID is used to read the file.
2. As a suggested practice, your workflow can clean up the properties file when it is no longer needed. If the file is a spool file, this action is not necessary because JES purges the file from the spool as part of normal processing.
3. If a spool file is used, and the file is purged before it is read by the Workflows task, the step is marked *Failed*.

Running a program in real time

A template step that runs a program in real time is called an *immediate execution step*.

To run an executable program (a REXX exec or UNIX shell script) in real time, include one of the following attributes on the submit as element (`<submitAs>`):

TSO-REXX

Run a REXX exec program in real time.

TSO-UNIX-REXX

Run a REXX exec program for the UNIX environment in real time.

TSO-UNIX-shell

Run a UNIX shell script in real time.

The program results are immediately available to the step owner.

Input parameters

To enable an executable program to receive input parameters from the step owner, include the element `<scriptParameters>` element with the following attributes:

<description>

Text description of the parameter, such as its intended use or recommended value.

<value>

Default value of the parameter.

Writing variables to a properties file

In your program, you might choose to create variables, and save them to a properties file. When the file is saved, the properties are available for use with the same workflow. For more information, see [“Creating a properties file” on page 1124](#).

Another workflow can access the properties file when the file is:

- Specified in the **Output File** field in the Workflows task UI.
- Read in as a workflow variable input file on creation of the workflow.

For a program that creates variables in a properties file, you can optionally include a prefix on the output variables. Here, you can specify a meaningful prefix that identifies a string as an output variable. To do so, add the element `<outputVariablesPrefix>` to the `<submitAs>` element.

In the following example, the string `outvar:` is applied to the names of any variables that are created by the step:

```
<outputVariablesPrefix needResolveConflicts="true">outvar:</outputVariablesPrefix>
```

To manage potential variable name conflicts, you can specify a default behavior by adding the attribute `<needResolveConflicts>` to the element `<outputVariablesPrefix>`. By default, the step owner is prompted to choose the appropriate variable value in the Workflows task.

For an array variable, variable name conflicts are handled differently. The step owner is not prompted to choose the appropriate variable value. Instead, variable name conflicts are handled by the behavior you specify on the `(loadOutputFileArrayValue)` attribute. Add this attribute to the element `<outputVariablesPrefix>` and set it to true or false, as follows:

- If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task.
- If set to false, the workflow uses the existing values from the Workflows task.

TSO/E address space for program execution

The executable program runs in a TSO/E address space on the z/OS host system. You can control how the TSO/E address space is started by including the following elements:

<regionSize>

Region size (in kilobytes) to be used for the address space. The valid range of values is 50000 - 2096128 (kilobytes). The Workflows task UI allows the step owner to specify a different region size, or use the supplied value. If no value is specified, the region size is 50000 kilobytes, by default.

<procName>

TSO logon procedure to be used for the address space. If no value is specified, the default is to use IZUFPROC, which is supplied by IBM as a cataloged procedure in proclib.

<timeout>

Time limit for an executable program. The valid range of values is 60 - 3600 (seconds). If no value is specified, the timeout value is 1800 seconds (30 minutes), by default.

Completion messages

On completion of program execution, standard TSO/E service messages are written to the script log. As the workflow author, you can supplement the TSO/E messages with your own customized messages to indicate successful completion or an error condition. To do so, add the following sub-elements to the `<submitAs>` element:

<successPattern>

Successful completion message. This element is required. You must specify one (and only one) regular expression for a successful program completion.

The format is:

```
<successPattern>success regular expression</successPattern>
```

<failedPattern>

Error message. This element is optional. You can omit this element or specify up to 100 different expressions for potential program errors.

The format is:

```
<failedPattern>failure regular expression</failedPattern>
```

Messages are limited in size, based on the program type, as follows:

TSO-REXX

The maximum length is 10000 lines.

TSO-UNIX-REXX

The maximum length is 255 characters; extra characters are truncated.

TSO-UNIX-shell

The maximum length is 255 characters; extra characters are truncated.

Temporary locations

Before running the program, z/OSMF saves the program to a temporary file. On completion of the program execution, z/OSMF deletes the temporary file. The location of the temporary file depends on the program type, as follows:

TSO-REXX

userID.IZUWFTSO.W, plus a randomly generated 7-digit number.

TSO-UNIX-REXX

/tmp/fileName-scriptRexxFile.rexx, where *fileName* is a randomly generated 32-digit number.

TSO-UNIX-shell

/tmp/fileName-scriptShellFile.sh, where *fileName* is a randomly generated 32-digit number.

Other related output files are saved to the following temporary locations:

- Variable output file is saved at /data/app/IZUWorkflows-workflowKey/outputFiles/stepName-outputfile
- Message output file is saved at /data/app/IZUWorkflows-workflowKey/scriptFile/stepName-scriptOutputMessageFile

Examples

In Figure 492 on page 1128, a template definition contains inline REXX commands for immediate processing. For more examples of running programs from a step, see file `workflow_sample_program_execution.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

```

<step name="TSO-TSO-REXX_Execution">
  <title>A step that runs a REXX exec program.</title>
  <description>In this step, an inline REXX exec is processed immediately on the host system.
    The processing is ended by a time-out condition.
  </description>
  <variableValue name="st_group" required="true"/>
  <variableValue name="st_user" required="true"/>
  <variableValue name="procNameVariable" required="true"/>
  <instructions substitution="false">This step outputs some variables and prints a few words.
  </instructions>
  <weight>1</weight>
  <skills>System Programmer</skills>
  <template>
    <inlineTemplate substitution="true">/* rexx */
    parse arg arg1
    SAY "this is a sample to submit TSO REXX script to run immediately"
    SAY "the first parameter is :" arg1
    SAY ${instance-st_user}
    SAY "prefix:st_group =" SYS123
    SAY "prefix:st_user =" USERS
    SAY "This execution will meets timeout."
    </inlineTemplate>
    <submitAs>TSO-REXX</submitAs>
    <successPattern>success.*</successPattern>
    <failedPattern>failed.*</failedPattern>
    <outputVariablesPrefix needResolveConflicts="true">prefix:</outputVariablesPrefix>
    <scriptParameters>
      <description>A simple parameter</description>
      <value>para1</value>
    </scriptParameters>
    <procName substitution="true">${instance-procNameVariable}</procName>
    <regionSize>50000</regionSize>
    <timeout>60</timeout>
    <saveAsUnixFile substitution="true">/u/${instance-st_user}/savedStuff/myScript.sh</saveAsUnixFile>
  </template>
</step>

```

Figure 492. This sample step submits a REXX exec for immediate processing.

Submitting a JCL job for batch processing

A template step that runs a program as a batch job is called a *batch execution step*. The submit as element (<submitAs>) indicates the type of program to be run.

To run an executable program as a batch job, such as a JCL job, a REXX exec, or a UNIX shell script, you include one of the following attributes on the submit as element (<submitAs>):

JCL

Submit a JCL job for batch processing on z/OS. The results are indicated in the job log.

TSO-REXX-JCL

Submit a JCL job that contains a REXX program. The program runs as a batch job on z/OS; the results are indicated in the job log.

shell-JCL

Submit a JCL job that contains a UNIX shell script. The program runs as a batch job on z/OS; the results are indicated in the job log.

Each program type is executed as a batch job by the Workflows task, which creates the necessary JCL and JOB statement, and displays the job output in the Workflows task. The size of any program to be run (JCL, REXX, or shell) is limited to 10000 lines of code.

In the following example, a template definition contains inline JCL that runs a TSO command:

```

<template>
  <inlineTemplate>//STEP1          EXEC  PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD    SYSOUT=A
//SYSTSIN  DD    *
LISTUSER IBMUSER
/*
  </inlineTemplate>
  <submitAs>JCL</submitAs>
</template>

```

Suppose that the first line of the JCL were to be placed on the next line after the beginning <inlineTemplate> tag. Doing so would inject a space into the JCL stream and cause a JCL error.

Similarly, if any of the lines of JCL were to be indented, a JCL error would occur. Therefore, unless the file or program is small, use `<fileTemplate>` instead, which identifies the path name of an external file that contains the template. For path name examples, see [“References to external files”](#) on page 1106.

In the following example, the file that is named `jcljob.txt` contains the JCL and exists in the same UNIX directory as the workflow definition file:

```
<template>
  <fileTemplate>
    jcljob.txt
  </fileTemplate>
  <submitAs>JCL</submitAs>
</template>
```

The Workflows task routes the job to another system in the sysplex, as determined by the system name that is chosen by the user when importing the workflow. Both JES2 and JES3 are supported. The JOB statement is entered by the Workflows task user separately, and applied to the job stream before it is submitted. Do not include a job card in your JCL template.

Observe the following considerations:

- A JCL template is submitted by the Workflows task after it includes the user-specified JOB statement and the appropriate JES routing statement, if the job is to be run on a different system in the sysplex.
- A REXX template is run by an IKJEFT01 job. This job creates a temporary data set to contain the REXX exec, and then executes the exec from that data set.
- A shell script is run by the BPXBATCH program. A temporary directory and file is created to contain the script. The script is run from this location and then the temporary directory and file are deleted.

Writing variables to a properties file

In your program, you might choose to create variables, and save them to a properties file. When the file is saved, the properties are available for use with the same workflow. For more information, see [“Creating a properties file”](#) on page 1124.

For a program that creates variables in a properties file, you can optionally include a prefix on the output variables. Here, you can specify a meaningful prefix that identifies a string as an output variable. To do so, add the element `<outputVariablesPrefix>` to the `<submitAs>` element.

In the following example, the string `outvar:` is applied to the names of any variables that are created by the step:

```
<outputVariablesPrefix needResolveConflicts="true">outvar:</outputVariablesPrefix>
```

To manage potential variable name conflicts, you can specify a default behavior by adding the attribute `<needResolveConflicts>` to the element `<outputVariablesPrefix>`. By default, the step owner is prompted to choose the appropriate variable value in the Workflows task.

For an array variable, variable name conflicts are handled differently. The step owner is not prompted to choose the appropriate variable value. Instead, variable name conflicts are handled by the behavior you specify on the `(loadOutputFileArrayValue)` attribute. Add this attribute to the element `<output>` and set it to true or false, as follows:

- If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task.
- If set to false, the workflow uses the existing values from the Workflows task.

Maximum return code

On the `<submitAs>` element, you can optionally indicate the maximum acceptable return code from the program on the `maxRc=` attribute.

When the user presses the **Close** button from the Status tab of the **Perform** page (which displays the job output), the Workflows task checks the job return code. If the return code is less than or equal to the

maxRc value, the Workflows task UI marks the step as complete. Otherwise, the Workflows task UI marks the step as failed.

JCL limits the return code value to the range 0 — 4095. If you do not specify a return code, the Workflows task uses a value of zero by default. BPXBATCH, which runs a shell script, further limits the return code to the range 0 — 15. If you are familiar with BPXBATCH, you might be aware that this service multiplies the script return code by 256 to derive a final return code value. The Workflows task divides the result by 256. Thus, you can code your workflow to be consistent with the value that is returned by the script.

Maximum record length

For a step that submits a job, you can use the `<maxLrec1>` element to specify the maximum record length, in bytes, for the input data for the job. This value is used when the step is performed automatically (`<autoEnable>` is set to true; see [“Automated steps” on page 1143](#)). If the step is performed manually, the user can optionally specify the maximum record length on the **Edit JCL** page in the Workflows task. If you omit the `<maxLrec1>` element, the default is 1024. For a job that uses the IEBUPDTE program to create or modify data sets, set this value to the minimum value of 80 to ensure that fixed-length records of 80 bytes or less can be processed.

Examples

For examples of submitting batch jobs from a step, see file `workflow_sample_wizards.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

Using variables in template steps

You can add customization capabilities to template steps through the use of step-specific variables and other techniques for designing steps, as described in this topic.

As the workflow author, you might want to:

- Specify certain variables as being able to be used in a JOB statement.
- Use the same JCL repeatedly over a number of steps, but with a different set of values substituted for each job submission.

This topic describes techniques that you can use to add flexibility to template steps, in the following topics:

- [“Variable substitution in the JOB statement” on page 1130](#)
- [“Predefined variables in a step” on page 1131](#)

Variable substitution in the JOB statement

As the workflow author, you might want to indicate that certain variables in the workflow can be used in a JOB statement. To make a variable available for user selection, include the `<exposeToUser>` element on the variable element tag. This element indicates that the variable is to be included among the user-selectable variables in the **List variables for substitution** window of the Workflows task.

With the `exposeToUser` element, you can identify a certain set of variables that can be used (through substitution) in the JOB statement when the step owner is performing the step. To use a variable, the step owner would have to modify the JOB statement to add the variable reference. The step owner can then view the substitution to see whether the JOB statement is as desired. The step owner can optionally designate the edited JOB statement be used for all steps in the workflow instance, but must understand the impact of doing so.

Suppose, for example, that you want to allow the user to select a notification user ID for the job. The system will send a message to this user ID when the job completes.

You can use the following technique to present the user with a selectable user ID FRED that is intended to be used with the NOTIFY= keyword parameter in the JOB statement. On the variable definition, include the exposeToUser element:

```
<variable name="notify-user">
<label>Notify user ID</label>
<abstract>This user ID can be selected for job notifications</abstract>
<description>Simple string variable.</description>
<exposeToUser>
<usage>Use this variable with the NOTIFY= parameter of the JOB statement
so that its value will designate who is notified when the job completes.</usage>
</exposeToUser>
<category>exposeToUser variables</category>
<string>
<default>FRED</default>
</string>
</variable>
```

In the workflow step, when the user displays the **Create JOB statement** page, the default JOB statement for the workflow is shown. For example, the following JOB statement is supplied with z/OSMF:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER
```

On the **Create JOB statement** page is the option **List variables for substitution**. If the user clicks this option, the Workflows task displays the variables in the workflow that include the exposeToUser element. The user can make JOB statement substitutions by manually copying these variables into the JOB statement.

In this example, the notify-user variable would be displayed in the **List variables for substitution** window. To use the notify-user variable for the notification user ID, the user can copy it into the JOB statement, as follows:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=${instance-notify-user}
```

On selecting the **View JOB Statement Substitutions** option, the user can display the substituted values to verify that the JOB statement is correct. In this example, the notification user ID is resolved to the value FRED:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=FRED
```

Predefined variables in a step

Suppose that your workflow contains a number of steps that run the same job repeatedly, except for some values in the body of the job that must change each time that the job is submitted. As an alternative to maintaining a slightly different copy of the JCL for each step, you can use the same job and call it from multiple steps. This technique requires that each step supply its own substitution values or *predefined variables* for each job submission. Doing so allows you as the workflow author to adjust the contents of the job dynamically to suit each step, based on the predefined variables.

To specify a predefined variable for a step, include the predefined variable (<predefinedVariable>) element on the step element tag, with the following element attribute:

name=

Name of the variable (a string).

A predefined variable is treated as a string substitution for the current step only. You can specify multiple predefined variables per step. To avoid overriding the variables that are defined for the workflow, use a unique name for the predefined variable.

REST steps

You can design a step to invoke a Representational State Transfer (REST) service. A step that calls a REST service is referred to as a *REST step*. This topic describes the use of REST steps and the Workflows schema elements that you can use to create them.

A typical use for a REST step would be to obtain values for your workflow and assign the values to variables defined in the workflow. Suppose, for example, that your workflow requires an IP address and a port number from a REST endpoint. A REST step could be used to invoke this REST endpoint and map the values from the REST endpoint response body to your IP address and port number variables.

An example of a REST step that obtains an IP address and port number is shown in [Figure 493 on page 1136](#).

Notes:

1. The REST service to be called must support the use of JSON as the content-type for the request body and the response body.
2. It is assumed that you understand the layout of the returned data and know where to obtain specific values from the JSON string for mapping to variables.
3. The returned values can be mapped to instance variables only. You cannot modify the values of global variables.

How the user interacts with a REST step

From the user's perspective, when a step definition includes the `<rest>` element, the Workflows task enables the **Perform** button on the step instructions page. When the user presses **Finish**, the perform wizard issues the REST request. The behavior of the wizard is further controlled through the sub-elements that you define within the `<rest>` element.

Elements of a REST step

In a workflow definition, a REST step is defined with the `<rest>` element and a number of sub-elements and attributes that provide details about the REST request. The details include the HTTP method, scheme name, URI path, query parameters, request body, expected status code, and actual status code. The `<rest>` element also defines any mapping properties that you can use to save the JSON properties (in the response body) to workflow variables.

[Table 577 on page 1132](#) provides a summary of the REST step elements.

Table 577. Summary of REST step elements				
Element Name	Description	Required or optional	Supports substitution	Default value
httpMethod	Indicates the HTTP method that is used for issuing the REST request. The following values are valid: <ul style="list-style-type: none">• GET• PUT• POST• DELETE.	Required	No	None

Table 577. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
schemeName	<p>Hypertext transfer protocol (HTTP) header for the request.</p> <p>The following values are valid::</p> <ul style="list-style-type: none"> • If you specify HTTP, you must also specify a host name for the receiving system. Optionally, you can specify a port number for the host name. The z/OSMF default port for HTTP requests is 80. • If you specify HTTPS, you must also specify the host name for the receiving system, and the user name and password that can be used to log in to the receiving system. Optionally, you can specify a port number for the host name. The z/OSMF default port for HTTPS requests is 443. • If you omit the scheme name and host name, the REST request is sent to the local z/OSMF server. <p>For more considerations, see the Table Notes.</p>	Optional	No	None.
hostname	<p>Host name or IP address of the system to which the REST request is directed. For example <code>www.ibm.com</code>.</p> <p>For more considerations, see the Table Notes.</p>	Optional	Yes	Host name that is defined for z/OSMF, which is "*" by default.
port	<p>Port number to use for the REST request.</p> <p>For more considerations, see the Table Notes.</p>	Optional	Yes	Port number that was defined for z/OSMF, which is "80" for HTTP requests and "443" for HTTPS requests.

Table 577. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
username	<p>z/OS user ID that allows the user to access the specified URI. This value is a user ID that is defined to your installation's z/OS security management facility (for example, RACF).</p> <ul style="list-style-type: none"> For an HTTPS request, the user name and password are required. These values are used to create a secure request to the specified URI. For an HTTP request, the user name and password are optional. These values are used for basic authentication only. 	Optional	Yes	None
password	<p>Password or pass phrase that is associated with the user name. For an HTTPS request, the user name and password are required.</p> <p>The Workflows task applies Base64 decoding to this value. Therefore:</p> <ul style="list-style-type: none"> To specify a password value on this element, encode the value by using Base64 encoding and specify the result here. For example, the string password would be specified as cGFzc3dvcmQ==. To use variable substitution for the password, ensure that the password variable is included in the workflow input variable file for the workflow, and is specified in Base64 encoded form. 	Optional	Yes	None
certificates		Optional	Yes	None
uriPath	URI path to use for the REST request.	Required	Yes	None
queryParameters	For a GET or POST request, this element contains the query parameters.	Optional	Yes	None
requestHeaders	This element contains the request header.	Optional	Yes	None
requestBody	For a PUT or POST request, this element contains the request body.	Optional	Yes	None
expectedStatusCode	The expected HTTP status code from the REST API request. If this value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a job template step returns a return code that is greater than the allowed maximum return code.	Required	No	None

Table 577. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
actualStatusCode	The actual HTTP status code that is received from the REST request. To obtain this value, map it to a workflow variable.	Optional	No	None
propertyMapping	The property from the REST response body that is mapped to a workflow variable. You can specify multiple propertyMapping elements in a REST step.	Optional	No	None

Table notes:

1. Scheme name and host name, and the optional port number, are used together. You must specify both scheme name and host name, or neither of them. Otherwise, the workflow fails validation when the user attempts to import the workflow definition into the Workflows task.
2. You must include the "mapTo" attribute on the <actualStatusCode> element. Set it to the name of the workflow variable that is to receive the actual status code value from the REST request. For example:

```
<actualStatusCode mapTo="status_code" />
```

Mapping the returned data to variables

You can assign the values that are returned from a GET request to variables that are defined in your workflow. To do so, include the <propertyMapping> element on the <rest> tag. This element identifies the required values in the JSON response body and saves the values to workflow variables. On the <propertyMapping> element, specify the value to be assigned on the "mapTo" attribute. You can specify multiple <propertyMapping> elements in a REST step.

In the following example, the workflow owner is defined in a property mapping.

```
<propertyMapping mapTo="workflow_owner">["name"]</propertyMapping>
```

If the response body is:

```
{ "name": "John" }
```

the workflow variable "workflow_owner" is assigned the value John.

Specifying substitution variables for REST step elements

You can specify substitution variables as values for the elements <hostname>, <port>, <uriPath>, <queryParameters>, and <requestBody>. Doing so allows you to set the values for these elements dynamically from the workflow variables.

For an element that contains variable substitution, do the following:

- Include the optional attribute "substitution" and set it to *true*. If not specified, the default is *false*. A value of *true* must be specified for the Workflows task to allow the variable substitution.
- Set the element value to *\${instance-WORKFLOW_VARIABLE_NAME}*, where *WORKFLOW_VARIABLE_NAME* is the workflow variable that is defined in the workflow.

A sample REST step is shown in [Figure 493 on page 1136](#). In this example, variable substitution is used in the <requestBody> element to allow dynamic substitution for the stackId and stackname properties.

```

<step name="get_ip_and_port_data">
  <title>Get values for workflow variables ip_addr and port from REST API call.</title>
  <description>Invoke RESTful API to get IP address and port number.</description>
  <prereqStep name="get_stack_data_for_rest_request"/>
  <instructions substitution="false"> Execute step to retrieve the IP address and port number
  from z/OS Communications Server.</instructions>
  <weight>10</weight>
  <skills>REST</skills>
  <rest>
    <httpMethod>POST</httpMethod>
    <uriPath>/zosmf/workflow/WorkflowManager/cloud/ipaddr/</uriPath>
    <requestBody substitution="true">
      {
        "name" : "GGAIPA",
        "description" : "IP address created by GGA",
        "ipaddr" : "1.1.1.2",
        "usageType" : "Internal",
        "workloadDeploymentGroup" : "/wdg/12345",
        "deploymentId" : "deployerA",
        "recoveryMethod" : "MANUAL_DISRUPTIVE",
        "hostName" : "mvstst1",
        "stack" :
          {
            "stackId" : "${instance-stack_id}",
            "stackname" : "${instance-stack_name}"
          },
        "boundTcpPorts" : [
          "4",
          "81"
        ]
      }
    </requestBody>
    <expectedStatusCode>201</expectedStatusCode>
    <actualStatusCode mapTo="status_code" />
    <propertyMapping mapTo="ip_address">["ipaddr"]</propertyMapping>
    <propertyMapping mapTo="port0">["boundPorts"][0]</propertyMapping>
    <propertyMapping mapTo="port1">["boundPorts"][1]</propertyMapping>
  </rest>
</step>

```

Figure 493. Sample REST step definition with substitution variables and property mapping variables

In this example, several returned values are mapped to workflow variables, as follows:

- `actualStatusCode` is mapped to the `"status_code"` variable.
- `"ip_address"` property is mapped to the `"ipaddr"` variable from the JSON object in the response body.
- `"port0"` property is mapped to the first element in an array of ports in the JSON object in the response body.
- `"port1"` property is mapped to the second element in an array of ports in the JSON object in the response body.

Example of a REST step

```
<step name="get_system_data_for_rest_request" optional="false">
  <title>Get System values now.</title>
  <description>Invoke REST API to get an unnamed array of variables.</description>
  <variableValue name="query_variant" />
  <variableValue name="body_sub" />
  <variableValue name="path_sub" />
  <instructions substitution="false">Execute this step to retrieve system values for this
workflow.
</instructions>
<weight>10</weight>
<skills>REST</skills>
<autoEnable>true</autoEnable>
<canMarkAsFailed>false</canMarkAsFailed>
<rest>
  <httpMethod>GET</httpMethod>
  <schemeName substitution="false">https</schemeName>
  <hostname substitution="false">www.mycompany.com</hostname>
  <uriPath substitution="true">/zosmf/${instance-path_sub}</uriPath>
  <queryParameters substitution="true">${instance-query_variant}</queryParameters>
  <requestBody substitution="true">${instance-body_sub}</requestBody>
  <expectedStatusCode>200</expectedStatusCode>
  <actualStatusCode mapTo="status_code"/>
  <propertyMapping mapTo="array2element3">["items"][0]</propertyMapping>
  <propertyMapping mapTo="array3element2">["items"][1]</propertyMapping>
  <username substitution="false">ibmuser</username>
  <password substitution="false">c3lzMQ==</password>
</rest>
</step>
```

Calling steps


Suppose that your workflow needs a function that is already performed by another workflow on your system. Rather than attempt to replicate the same function in your workflow, you can add a step to start the other workflow as needed. A workflow that starts another workflow is the *calling workflow*. A workflow that is started by another workflow is the *called workflow*. On completion, the called workflow returns control to the calling workflow.

From the point of view of the calling workflow, the called workflow is simply another "step" to be performed. You define the called workflow in the workflow definition file for the calling workflow, on the step element (<step>), through a set of elements and attributes.

With the ability to call one workflow from another, you can follow a modular approach to designing workflows. You can include a common, frequently performed, action in one workflow and allow other workflows to call that workflow, as needed. By including one or more called workflows in your design, you can maximize reuse of your code by using "common workflows," and effectively, integrate several workflows together.

To make a workflow eligible to be called by another workflow, include the *isCallable* attribute on the workflow metadata element <workflowInfo>, as described in [“Callable workflows” on page 1111](#).

How the user interacts with a calling step

In the Workflows task user interface, a step that starts a called workflow is indicated to the user with the following icon in the **Workflow Steps** page: 

In the Workflows task, when the user reaches a step that calls another workflow, z/OSMF determines whether an instance of the called workflow is already active on the system. If so, the Workflows task redirects focus to the called workflow so that the step owner can complete it. If the called workflow is already completed, the Workflows task displays the Workflows Steps table, showing the step that started the called workflow as marked complete.

Otherwise, if the called workflow is not already created, the Workflows task opens to the **Create Workflow** page so that the user can create the workflow. The user can supply a name for the workflow on the **Create Workflow** page, or accept the default workflow name.

Coordinating workflow-to-workflow actions

The Workflows schema metadata attributes *scope* and *isCallable* can be useful for coordinating workflow-to-workflow actions across a system or sysplex. By setting these attributes in various combinations, you can achieve the following effects:

- Restrict the number of instances of a workflow to one instance (a *singleton*).
- Always attempt to use an existing instance of a workflow in response to a calling workflow. Or, always cause a new instance of a workflow to be created, even if an instance exists already.
- Limit the use of a callable workflow to the same system or sysplex.

The *scope* and *isCallable* attributes are specified on the workflow metadata element (<workflowInfo>), as described in [“Callable workflows” on page 1111](#).

Nine combinations of *scope* and *isCallable* are possible, as shown in [Table 578 on page 1138](#).

Table 578. Use <i>scope</i> and <i>isCallable</i> to coordinate workflow-to-workflow actions		
scope value	isCallable value	Effect on the called workflow
system	system	Workflow is limited to one instance per system, and can be called on that system only. If an instance does not exist, an instance is created.
sysplex	system	Workflow is limited to one instance per sysplex, and can be called from the same system as the calling workflow.
none	system	A new instance is always created. The instance can be called from the same system as the calling workflow.
system	sysplex	Workflow is limited to one instance per system, and can be called from any system in the sysplex. If an instance does not exist, an instance is created. For an automated workflow, if the calling step is performed automatically, the called workflow is searched for only on the calling system. If an instance is found, it is used; otherwise a new instance is created on the calling system.
sysplex	sysplex	Workflow is limited to one instance per sysplex, and can be called from any system in the sysplex. If an instance does not exist, an instance is created.
none	sysplex	A new instance is always created. The instance can be called from any system in the sysplex.
system	none (omitted)	Workflow is not callable.
sysplex	none (omitted)	Workflow is not callable.
none	none (omitted)	Workflow is not callable.

How workflow access type is handled

When a workflow calls another workflow for processing, z/OSMF changes the access type for the called workflow to match the calling workflow. This processing ensures that the requested access type is applied consistently to both of the workflows in a calling relationship.

However, this processing is not performed when the called workflow is limited to one active instance in the system or sysplex.

Designing a step to call another workflow

To have a step call another workflow, specify the step element (<step>) with the elements for defining a called workflow. The major step elements for defining a called workflow in your workflow definition file are described in [Table 579 on page 1139](#).

Table 579. Major step elements for defining a called workflow in your workflow definition file		
Element name	Description	Required or optional
variableMapping	Used to transfer instance variable values between the called workflow and calling workflow, and specify options for sharing variables. More information about this element and its sub-elements is provided in “Sharing variables between the calling workflow and called workflow” on page 1140.	Optional.
callingStepWeight	Specifies the relative difficulty of the step compared to other steps within this workflow (a positive integer value 1 – 1000). The Workflows task uses this value in the calculation of the percentage-complete value that is displayed for the calling workflow. The scale is arbitrary; specify it at your discretion. Consider the difficulty of the called workflow as single step among the other steps in the calling workflow.	Required.
callingStepSkills	Specifies a suggested skills category for performing the step, such as "Security administration" or "Network administration." The Workflows task displays this value in the step table for a workflow. This value is free-form; specify it at your discretion.	Optional.
callingStepAutoEnable	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>callingStepAutoEnable</i> is not specified, the default is <i>false</i> . More information about designing steps to run automatically is provided in “Automated steps” on page 1143	Optional.
canCallingStepMarkAsFailed	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. When set to <i>true</i> , the Review Instructions page in the Step Perform wizard includes the option to allow the step owner to mark a step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user. If <i>canCallingStepMarkAsFailed</i> is not specified, the default is <i>false</i> .	Optional.
calledWorkflowDefinitionFile	Specifies the external file that contains the workflow definition for the called workflow. Provide the absolute (fully qualified) path name, or a relative path name (that is, relative to the location of the calling workflow). This element is optional; it is used only if z/OSMF must create a new instance of the called workflow on the system. For a relative path, the path must begin with <i>./</i> or <i>../</i> . After this notation, all subsequent instances of <i>./</i> or <i>../</i> in the path will be resolved. An example of using a relative path is shown in Figure 494 on page 1143 .	Optional.
calledWorkflowDescription	Specifies a short description of the called workflow. The Workflows task displays this text on the Create Workflow page, when it prompts the user for the workflow definition file.	Required.
calledWorkflowID	The name of the workflow to be called. The combination of <i>calledWorkflowID</i> and <i>calledWorkflowVersion</i> must be unique within the Workflows task.	A selection is required: Either <i>calledWorkflowID</i> or <i>calledWorkflowMD5</i> .

Table 579. Major step elements for defining a called workflow in your workflow definition file (continued)

Element name	Description	Required or optional
calledWorkflowVersion	The version of the definition file that is used to create the called workflow. The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task. The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify any portion of the workflow definition file, including changes to any sub-files or referenced files. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.	Optional.
calledWorkflowMD5	An MD5 encrypted value (a 128-bit hash value) that you can use to identify the called workflow.	A selection is required: Either calledWorkflowID or calledWorkflowMD5.

Identifying the called workflow

To start a called workflow, the calling step must identify which workflow is to be called. The z/OSMF schema provides two methods for you to reference the called workflow. Use either of the following approaches:

- Specify the workflow ID of the called workflow on the workflow ID element (<calledWorkflowID>). You can further qualify this specification by optionally including the version of the workflow definition of the called workflow on the element (<calledWorkflowVersion>). The version is typically updated by the workflow author whenever any portion of the workflow definition file is changed.

The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.

- Specify the called workflow MD5 element (<calledWorkflowMD5>). This element specifies a 128-bit hash value that can be used to identify the called workflow. You can specify this element in place of the workflow ID and version elements.

Note: No more than one level of nesting of called workflows is permitted in a workflow-to-workflow relationship. Thus, the specified workflow definition cannot contain a step that calls another workflow.

Sharing variables between the calling workflow and called workflow

It is possible to share variables between the calling workflow and the called workflow. Any variables that are defined to either workflow can be shared by using the element <variableMapping>.

This element consists of two sub-elements and their associated attributes, as follows:

- Use the element <fromCallingToCalled> to describe the variable values that are to be transferred from the calling workflow to the called workflow.
- Use the element <fromCalledToCalling> to describe the variable values that are to be transferred from the called workflow to the calling workflow. To handle variable conflicts, you can optionally include the attribute `override=` to specify whether the called workflow variables take precedence over the calling workflow variables. The default is `override=false`.

On each element, you can optionally specify the following sub-elements and attributes:

regExpression

Regular expression. Use this attribute to filter on variable names with one or more wildcard characters. For example, to select all variables prefixed with "setting," you can specify:

```
<regExpression>^setting.*$</regExpression>
```

variableName

Name of the variable. Use this element to identify the variable that is to be shared with the target workflow. The variable is also saved in the Workflows task variable pool.

To map this variable to a specific variable in the target workflow, include the attribute `mapTo=` on the element `variableName` and set it to the name of the target variable. The behavior of the attribute `mapTo=` on the element `variableName` depends on which element is used to pass variables, as follows:

- When specified on the element `<fromCallingToCalled>`, the variables are mapped to the target variables only when a new instance of the called workflow is created in response to the calling step.
- When specified on the element `<fromCalledToCalling>`, the variables are mapped to the target variables on completion of the called workflow.

In some cases, you might not know which of your workflow variables are needed by the called workflow. Suppose, for example, that the called workflow is supplied by a different workflow provider. Creating variable mappings in such cases is not possible. As an alternative to creating variable mappings, you can share your instance variables implicitly with a called workflow by making the variables publicly visible. z/OSMF makes all of the calling workflow's publicly visible instance variables available to the called workflow. These variables are referred to as *caller scope* variables. They are shared only with the called workflow. For more information, see [“Caller scope variables” on page 1169](#).

Sharing the account information and JOB statement with the called workflow

If the calling workflow contains account information and a JOB statement, this information is propagated to the called workflow. On creation, the called workflow can use the passed information if it submits a job to run. To receive the JCL information, the called workflow must have a workflow scope of NONE.

The passed information is:

accountInfo

Account information to use in the JCL JOB statement.

jobStatement

JOB statement JCL that is used in the job.

If this JCL information is present in the calling workflow, z/OSMF stores it when the called workflow is created. If the called workflow submits a job to run, z/OSMF applies the account information and JOB statement from the calling workflow to the job that is submitted by the called workflow.

If no account information and JOB statement can be obtained from the calling workflow, the called workflow uses the default JOB statement that is supplied with z/OSMF:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,  
MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER
```

Example of how a called workflow is defined in a step

As an example, assume that your workflow includes a step ("Define User"), which is used to define a user ID and security group to the system security product. Usually, to verify that this setup is done correctly, users would run another workflow. In this example, you add a step to start the other workflow directly as a called workflow. When the step owner selects this step to be performed, the Workflows task displays the called workflow so that the step owner can perform it.

Further assume that a number of variables are shared between the workflows by using the element `<variableMapping>`. The step that calls the workflow (the calling step) passes a number of variables on the element `<fromCallingToCalled>`. Similarly, the called workflow passes a number of variables to the called workflow, on the element `<fromCalledToCalling>`.

The definition for a called workflow might be coded as shown in [Figure 494 on page 1143](#).

For illustrative purposes, the example in [Figure 494 on page 1143](#) shows various methods for sharing variables between workflows, as follows:

- On the element `<fromCallingToCalled>`, the step that calls the workflow (the calling step) passes variables in the following ways:
 - To pass all variables with "setting" as the variable name prefix, the calling step specifies the element `<regExpression>`, as follows:

```
<regExpression>^setting.*$/regExpression>
```

- To pass the value of the variable that is called `st_user` to the called workflow, the calling step specifies the variable, as follows:

```
<variableName>st_user</variableName>
```

- If an instance of the called workflow is not already created, z/OSMF creates one in response to the called workflow. If so, the element `<fromCallingToCalled>` ensures that the new called workflow inherits the variables from the calling workflow. In this example, the value of the variable `st_uid` is passed to the calling workflow, and overlays the existing value of the variable that is named `st_group` because the attribute `mapTo=` is included on the element `<variableName>`, as follows:

```
<variableName mapTo="st_group">st_uid</variableName>
```

- On the element `<fromCalledToCalling>`, the called workflow shares a number of variables with the calling workflow. Any variables to be copied back to the calling workflow are performed on completion of the called workflow. Here, the override attribute is included so that the called workflow's variables override those of the calling workflow:

- To pass all variables with "set" as the variable name prefix, the element `<regExpression>` is specified, as follows:

```
<regExpression>^set.*$/regExpression>
```

- To pass the value of variable that is called `st_uid` to the calling workflow, and overlay its existing value for the variable that is named `st_gid`, the `mapTo=` attribute is included on the `variableName` element, as follows:

```
<variableName mapTo = "st_gid">st_uid</variableName>
```

- To pass the variable that is called `st_user` to the calling workflow, the variable is specified, as follows:

```
<variableName>st_user</variableName>
```

In [Figure 494 on page 1143](#), a relative path is provided for the location of the called workflow definition file on the element `<calledWorkflowDefinitionFile>`. By using a relative path instead of an absolute path, you provide the location of the file in relation to the calling workflow. To be found during processing, the file must reside in the same file system as the calling workflow.

In the example, the location of the called workflow is specified as:

```
<calledWorkflowDefinitionFile>..\samples\workflow_sample_wizards.xml</calledWorkflowDefinitionFile>
```

If the calling workflow is located in the `\usr\lpp\zosmf` directory, the called workflow location resolves to `\usr\lpp\zosmf\samples\workflow_sample_wizards.xml`.

```

<step name="Define User">
  <title>Ensure that the user ID and group are created.</title>
  <description>
    This step verifies that the required user ID and security
    group are created. This step invokes another workflow (a called workflow),
    which is identified here based on the workflow ID and version.
    Alternatively, we could have identified the called workflow using its
    MD5 hash value.
  </description>

  <variableMapping>
    <!-- Variables to share with the called workflow. -->
    <fromCallingToCalled>
      <!-- Use a regular expression to filter the variables. -->
      <regExpression>^settin.*$/regExpression>

      <!-- The following line copies the value of st_uid to the variable st_group. -->
      <variableName mapTo = "st_group">st_uid</variableName>

      <!-- The following line copies the value of st_user to the called workflow,
      if no st_user variable definition already exists in the called
      workflow.
      This value also will be saved in the Workflows task variable pool.-->
      <variableName>st_user</variableName>
    </fromCallingToCalled>

    <!-- Variables to share with the calling workflow. Here, the override attribute
    is set to true, so that the called workflow's variable values will override
    those of the calling workflow. -->
    <fromCalledToCalling override= "true">
      <regExpression>^set.*$/regExpression>
      <variableName mapTo = "st_gid">st_uid</variableName>
      <variableName>st_user</variableName>
    </fromCalledToCalling>
  </variableMapping>

  <callingStepWeight>10</callingStepWeight>
  <callingStepSkills>System Programmer</callingStepSkills>

  <calledWorkflowDefinitionFile>.\..\samples\workflow_sample_wizards.xml
  </calledWorkflowDefinitionFile>

  <calledWorkflowDescription>This called workflow is used to help verify that the user
  and group are created successfully.</calledWorkflowDescription>

  <calledWorkflowID>workflow.sample.wizards</calledWorkflowID>

  <calledWorkflowVersion>1.0</calledWorkflowVersion>
</step>

```

Figure 494. Example: Defining a called workflow on the step element tag.

Automated steps

A workflow might have steps that can be performed without the need for user interaction, such as a job that can be submitted without user input. If so, you can designate the step as an *automated step* in the workflow definition file. Doing so instructs the Workflows task to run the step automatically, as soon as any prerequisite steps in the workflow are completed. By including one or more automated steps in a workflow, you help to simplify the user experience.

In the Workflows task user interface, automated steps are indicated to users in the following ways:

- In the *Workflow Steps* table, the column Automated indicates whether a step is automated, based on how the step is defined in the workflow definition.
- In the Workflows task main page, when an automated step is performed, the workflow status is indicted as *Automation in Progress*.

A workflow can consist of both automated steps and non-automated (manually performed) steps.

How the user interacts with an automated step

To perform an automated step, the user selects the **Perform** action in the *Workflow Steps* table, as is done for manual steps. For an automated step, the Workflows task presents the user with a dialog window to confirm whether the step and any subsequent automated steps are performed automatically. Alternatively, the user can choose to perform the step manually, by selecting an option in the dialog window that is called **Manually perform the selected step**.

When started, an automated step—or series of automated steps—can run to completion or until stopped by another condition, such as a user stop request or a step error. Specifically, a workflow with automated steps can run until one of the following conditions occurs:

- Completion of all subsequent steps.
- Processing reaches an automated step for which one or more required variables are not satisfied.
- Processing reaches a non-automated step in the sequence of steps.
- Processing reaches an automated step that is not currently eligible for automatic processing. That is, the step is *Unassigned*, *Assigned*, *Not Ready*, or *Submitted*.
- Processing is stopped through a user request.
- An error is encountered.

A workflow that is comprised entirely of automated steps can run to completion without user intervention.

Tracking the progress of automated steps

For workflows that contain automated steps, z/OSMF creates notifications and history entries to inform step owners of the automation progress.

At the completion of an automated step or a sequence of automated steps, z/OSMF creates a notification to inform the step owner of the step status. If processing reaches a manual step that requires user interaction before the workflow can continue, z/OSMF creates a notification for the step owner to prompt for action. Similarly, if an automated step is stopped or fails for any reason, z/OSMF sends a notification to the step owner. In z/OSMF, users can access notifications through the Notifications task.

During the processing of an automated step, z/OSMF updates the workflow history to indicate the key checkpoints in the workflow progress, such as:

- Completion of the automated step
- Completion of all automated steps in the workflow
- Automation is started or stopped through user request (and by whom)
- An error is encountered during the processing of an automated step.

Automation progress is not displayed (in terms of step completion check marks) until the user refreshes the display.

Users of the Workflows task can view the details of the step status in the Workflow History table.

Design considerations for automated steps

Consider a step to be eligible for automation if it requires no user input at all. Or, if all of the required inputs can be supplied to the workflow at creation time, in the form of a workflow variable input file.

When you code the step element, you can specify whether the step is automated (that is, can be performed automatically by the Workflows) by including the `autoEnable` attribute on the step element (`<step>`). Set this attribute to true or false, as needed. By default, the `autoEnable` attribute is false. [Figure 495 on page 1145](#) shows an example.

```

<step name="Sample_Automated_Step" optional="true">
<title>This is a very simple JCL job</title>
<description>This optional step submits an empty job using IEFBR14.</description>
<instructions>This step is performed automatically.</instructions>
<weight>1</weight>
<skills>Submit a job to run on z/OS</skills>
<autoEnable>true</autoEnable>
<template>
  <inlineTemplate>//STEP1      EXEC  PGM=IEFBR14
//SYSEXEC DD DUMMY
//* PRINT DD SYSOUT=A
/*
  </inlineTemplate>
  <submitAs>JCL</submitAs>
</template>
</step>

```

Figure 495. You can designate a step as automated by adding the `autoEnable` element to the `<step>` element tag.

For any automated steps that you provide, it is recommended that you use the description tag to provide the user with enough information to understand the implications of allowing the step to run.

When automated steps are ordered consecutively in a workflow, a request to run the first automated step begins a process in which each subsequent automated step can run to completion, or until one of the steps encounters a condition that stops the processing of steps. For this reason, it is recommended that you group automated steps in the workflow definition file together to take full advantage of this cascading behavior.

If you do not rely on automated steps to complete in a certain sequence, consider using parallel processing for a potentially faster completion time for your workflow. For details, see [“Enabling automated steps for parallel processing”](#) on page 1146.

Using the suspend element to control automation

As the workflow author, you can force automation to stop before a particular step is run. You might do so, for example, to pause automation so that someone can perform an action outside of the workflow before the workflow resumes automated processing.

To force a stop in the sequence of automated steps, include the suspend element (`<suspend>`) in the step definition. When automation processing reaches a step that includes the suspend element, automation stops at the step. Here, z/OSMF can send an email notification to one or more recipients that you specify, such as a person who should be prompted to take action. You might, for example, need to direct someone to configure a resource, or collect some information that the workflow owner needs to proceed with the workflow. Or, you might simply need to notify someone that a created object is available for use. The properties of the email are described in [Table 580 on page 1146](#).

To resume the workflow, a user can choose to manually run the suspended step, or specify that automation is to resume from the step that contains the suspend element. In the later case, automation ignores the suspend element and begins automation from the selected step.

On the suspend element, you can specify the settings to create the notification email. [Table 580 on page 1146](#) describes these sub-elements of the suspend element.

Table 580. Information to specify for notifying a user about a suspended step.

Element	Description
toRecipients	<p>Specify the email addresses of the persons to be notified of the suspended step. To specify more than one recipient, enter each address, separated by commas or spaces.</p> <p>It is possible to use variable substitution in this field.</p> <p>In addition to any users that you specify on the <toRecipients> element:</p> <ul style="list-style-type: none"> • For a configuration or general workflow, z/OSMF always sends a notification to the workflow owner. • For a provisioning workflow, z/OSMF sends a notification to the domain administrator, if this user identity is defined. Otherwise, z/OSMF notifies the workflow owner.
subject	<p>Specify a brief, meaningful subject for the notification email. If you omit this value, the email subject is set to no subject by default.</p> <p>It is possible to use variable substitution in this field.</p>
content	<p>Specify the text of the message that you want to send to the recipient. If you omit this value, the email content is set to no content by default.</p> <p>It is possible to use variable substitution in this field.</p>

Figure 496 on page 1146 shows an example of how the suspend element can be coded within a step definition.

```
<description>Submit an empty JCL job to JES using IEFBR14.</description>
<suspend>
<toRecipients>neelash@my.company.com, conner@my.company.com</toRecipients>
<subject>Automation suspended</subject>
<content>These are the instructions to resume automation and contact the
domain administrators</content>
</suspend>
<instructions substitution="false">This is a very simple JCL job submission that is
performed
by using IEFBR14.</instructions>
```

Figure 496. Example of a suspend step

In Figure 496 on page 1146, the email notification:

- Is sent to the recipients neelash@my.company.com and conner@my.company.com
- Has the subject "Automation suspended"
- Contains the following text: "These are the instructions to resume automation and contact the domain administrators."

Note: For a suspended step, the *Step Properties* page in the Workflows task includes a tabbed area called **Suspend Information**. This area displays the email settings that are specified on the sub-elements of the suspend element.

Enabling automated steps for parallel processing

As the workflow author, you can request that automated steps be run in parallel (concurrently), rather than sequentially. A workflow with steps that can be run in parallel is called a *parallel-steps workflow*.

With parallel processing, a workflow can take less time to complete. However, the steps might complete in an unexpected sequence. Consider using parallel processing when you do not rely on automated steps to complete in a certain sequence.

To use parallel processing, include the attribute `parallelSteps=true` in the workflow metadata. Otherwise, if this attribute is set to `false` or omitted, automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

When a parallel-steps workflow is started, the Workflows task locates the automation ready steps and attempts to run them concurrently.

A step is considered to be *automation ready* when it is:

- Enabled for automation. In the workflow definition file, the attribute `autoEnable=true` is specified on the step element (`<step>`).
- In an eligible state: *ready*, *in-progress*, or *failed*. For a failed step, the Workflows task performs the step again.

In a parallel-steps workflow, the failure of an automated step does not stop automation processing for the other automated steps. Processing continues until all of the automated steps are completed or failed, or a condition occurs that stops automation processing, such as a user stopping automation by using the **Stop Automation** action in the Workflows task.

Note: The ability to suspend step processing is mutually exclusive with the ability to run steps in parallel. Therefore, if a workflow includes the suspend element (`<suspend>`) in the step definition, it is not eligible for parallel processing. The Workflows task enforces this restriction. An attempt to start a workflow that contains both the suspend element (`<suspend>`) and the attribute `parallelSteps=true` results in an error.

Making a step conditional

A *conditional step* is available to be performed when a logical condition is satisfied on the z/OS system or in the Workflows task. A conditional step might become *Ready* (eligible to be performed), for example, if a job run by another step ends with a particular return code. A conditional step remains *Not Ready* (unavailable to be performed) as long as the condition is not satisfied.

Understand that a conditional step, which depends on a logical condition, is different than a *dependent* step, which depends on a particular step being completed, to satisfy a prerequisite.

In the Workflows task user interface, conditional steps:

- Are indicated to users in the **Details** tab on the *Step Properties* page.
- Are shown in the *Not Ready* state until the condition is true (satisfied) — even when the prerequisite steps, if any, are complete.

A conditional step becomes ready for performing only when a specific condition is satisfied in the current step or a preceding step. Thus, the expression being tested and a text description are required sub-elements of the condition element.

Target states

Optionally, you can specify a desired *target state* for a conditional step. The target state specifies the state the step is to assume when the condition is true. Typically, the target state is *Ready*, which is the default value, if you choose to omit this sub-element.

The following target states are valid:

- Ready
- Skipped
- Complete.

A conditional step remains unavailable to be performed as long as the condition is not satisfied. This rule applies even when `prereqStep` or `prereqTargetStateSet` are also defined for the step. Also, a `targetStateSet` with a condition is evaluated only when the condition is true.

Types of conditional expressions

The following types of conditional expressions are supported:

- Expressions using logical operators AND (&) and OR (|). For example:

```
${step1.returnValue} == "0000" || (${step2.returnValue} == "0000" && ${step2.stepOwner} == "IBMUUSER")
```

- Expressions based on ternary operators. For example:

```
condition ? value_if_true : value_if_false
```

- Mathematical functions. For example:

```
Math.max(${step1.returnValue} , ${step2.returnValue} ) > 0
```

Design considerations for conditional steps

Observe the following design considerations for conditional steps:

- A conditional step must be a leaf step (a step with no substeps). A parent step cannot be a conditional step.
- When coding the step element, specify whether the step is conditional by including the `condition` attribute on the step element (`<step>`). Also, specify both the expression being tested for (typically a mathematical or logical expression) and a text description of the condition. Both the expression and its description are displayed to the end user in the **Details** tab on the *Step Properties* page.
- A conditional expression can refer only to preceding steps in the workflow.
- You can include workflow input variables in conditional expressions. Doing so allows conditional steps to resolve to true or false, based on installation-specific conditions.
- You can use the following step attributes in conditional expressions: `<stepState>` and `<returnValue>`.
- `<returnValue>` is a string type attribute; you cannot use it in a mathematical comparison. To compare a return code with a second return code or another numerical value, such as zero (0), you can write the condition expression like this: `${step2.returnValue}) > "0000"`. Represent the return code string with four characters, for example "0000" or "0008".

Example

As an example, assume that Step 3 should not be performed unless Step 1 and Step 2 complete with a return code zero. Here, the XML for Step 3 could be coded as follows:

```

<Step name=Step3">
  <title>A conditional step based on return code</title>
  <description>This conditional step is not ready until
                the two preceding steps complete with RC 0
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression>${step1.returnCode} == "0000" || (${step2.returnCode} == "0000"
    </expression>
    <description>This step requires that Steps 1 and 2 have
                  completed successfully.
  </description>
</condition>
<targetState>Ready</targetState>

```

Figure 497. You can designate a step as conditional by adding the condition element to the <step> element tag.

The previous example can be expanded to include a condition, based on a variable value. In [Figure 498 on page 1149](#), Step 3 is not performed unless Step 1 and Step 2 complete with a return code zero and the variable `${instance-st_user}` is IBMUSER.

```

<Step name=Step3">
  <title>A conditional step based on return code and user ID</title>
  <description>This conditional step is not ready unless
                the two preceding steps complete with RC 0
                variable st_user value is IBMUSER
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression><![CDATA[${step1.returnCode} == "0000" &&
    </expression>
    <description>This conditional step is not ready unless the two
                  preceding steps complete with RC 0 and the variable
                  st_user value is IBMUSER.</description>
  </condition>
  <targetState>Ready</targetState>
  :
</step>

```

Figure 498. You can use variable values in the condition to be satisfied.

Note that a variable reference can contain an underscore, for example: `${instance_st_user} == "IBMUSER"` or a hyphen, for example: `${instance-st_user} == "IBMUSER"`.

runAsUser identity for a step

For workflow automation, you can specify the user ID under which a particular step is to be performed by including the element `runAsUser` (<runAsUser>) on the step element. The value that you specify on the `runAsUser` element is considered to be the *runAsUser ID* for the step.

When a `runAsUser` is not specified for a step, the step is performed under the step owner user ID.

The `runAsUser` ID element is intended for use with an automated workflow. The `runAsUser` ID element is not applicable in non-automated cases; its use can result in an error that prevents the step from being performed. For the intended use cases, see [“How a runAsUser ID is used in workflow automation” on page 1150](#).

If you specify the `runAsUser` element, you must ensure that the user ID that you specify is (or will resolve to) a valid z/OS user ID. The user ID can be lower case, upper case, or mixed case.

How a runAsUser ID is used in workflow automation

During workflow automation, a step is switched to the runAsUser ID for the operations that are shown in Table 581 on page 1150.

Table 581. How a runAsUser ID is used in workflow processing		
Step type	Automated operation	How the runAsUser ID is used
Template step	Step submits a JCL job (a batch execution step).	Batch job runs under the runAsUser ID.
Template step	Step runs a program in real time (an immediate execution step).	Program runs under the runAsUser ID.
Template step	Step copies an output file to its own storage.	Read action is performed under the runAsUser ID.
Template step that includes the saveAsDataset element.	Step writes the template contents to a data set.	Save as a data set action is performed under the runAsUser ID.
Template step that includes the saveAsUnixFile element.	Step writes the template contents to a UNIX file.	Save as a UNIX file action is performed under the runAsUser ID.
REST step	Step calls a REST service.	REST service is invoked under the runAsUser ID.

For other step types and operations, the runAsUser ID is not applicable. Specifically, the runAsUser ID is not used for the following step types:

- Instructions steps
- Calling Steps

If a workflow definition that does not originate from z/OS Management Services Catalog or IBM Cloud Provisioning and Management for z/OS includes runAsUser steps, a signature is required before a step can be performed. For more information, see [“Specifying signers for a step” on page 1150](#).

Signers and signatures cannot be specified in a workflow definition that originates from z/OS Management Services Catalog or IBM Cloud Provisioning and Management for z/OS. For these workflow definitions, see [“Specifying approvers for a step” on page 1152](#).

Specifying signers for a step

The runAsUser element is typically used with the stepSignature element, as follows:

- The runAsUser element (<runAsUser>) specifies the user identity under which the step is to be performed.
- The stepSignature element (<stepSignature>) specifies signer element and signature element. The use of a stepSignature element requires that the runAsUser element be specified. The element (<stepSignature>) and the element (<approver>) are mutually exclusive.
- The signer element (<signer>) specifies who must sign the step legally before the step can be performed. You must ensure that the signer that you specify is a valid z/OS user ID or valid z/OS group ID. A user ID or a list of user IDs separated by blanks. At least one user ID must sign the step before it is performed on behalf of the user ID that is specified with the runAsUser element. To specify multiple required signers, use multiple signer elements (up to 12). The signer element is optional. If it is specified, the runAsUser element is required.

- The signature element (<signature>) specifies the signature which is generated after the signer signs for the step in Workflow Editor task. The use of the signature element requires that both the runAsUser and signer elements be specified. It is omitted if the signer element is not specified.
- The version element (<version>) specifies the version of the signature.

Various combinations of signers are supported for the runAsUser element. In the examples that follow, assume that a workflow includes a step that is performed under an authorized user identity, which is represented by the runAsUser ID liqiliqi.

- In the following example, either ywzhao or lbtliu must sign for the step:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
  <stepSignature>
    <sign>
      <signer>ywzhao lbtliu</signer>
    </sign>
  </stepSignature>
```

After either ywzhao or lbtliu sign the step, the result is as follows:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
  <stepSignature>
    <sign>
      <signer>ywzhao lbtliu</signer>
      <signature version="1.0">*****</signature>
    </sign>
  </stepSignature>
```

- In the following example, both ywzhao and lbtliu must sign for the step:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
  <stepSignature>
    <sign>
      <signer>ywzhao</signer>
    </sign>
    <sign>
      <signer>lbtliu</signer>
    </sign>
  </stepSignature>
```

After both ywzhao and lbtliu sign the step, the result is as follows:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
<stepSignature>
  <sign>
    <signer>ywzhao</signer>
    <signature version="1.0">*****</signature>
  </sign>
  <sign>
    <signer>lbtliu</signer>
    <signature version="1.0">*****</signature>
  </sign>
</stepSignature>
```

- In the following example, gqdeng and either ywzhao or lbtliu must sign the step:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
<stepSignature>
  <sign>
    <signer>gqdeng</signer>
  </sign>
  <sign>
    <signer>ywzhao lbtliu</signer>
  </sign>
</stepSignature>
```

After gqdeng and either ywzhao or lbtliu sign the step, the result is as follows:

```
<runAsUser substitution="false">liqiliqi</runAsUser>
<stepSignature>
  <sign>
    <signer>gqdeng</signer>
    <signature version="1.0">*****</signature>
```

```

        </sign>
        <sign>
            <signer>yzhao lbtliu</signer>
            <signature version="1.0">*****</signature>
        </sign>
    </stepSignature>

```

Important: When using stepSignature (<signature>) element, the runAsUser (<runAsUser>) element does not support substitution. The substitution in runAsUser element always be false.

At workflow creation time, z/OSMF checks the runAsUser ID. The ID should be a z/OS user ID that has permissions to the z/OSMF SAF profile prefix profile in the APPL class. If these checks fail, the workflow cannot be created.

Specifying approvers for a step

In z/OS Cloud Provisioning, the runAsUser element is typically used with the approver element, as follows:

- The runAsUser element (<runAsUser>) specifies the user identity under which the step is to be performed.
- The approver element (<approver>) specifies who must grant approval before the step can be performed. Up to 12 approvers can be specified for a step. The use of an approver element requires that the runAsUser element be specified.

Various combinations of approvals are supported for the runAsUser element. In the examples that follow, assume that a workflow includes a step that is performed under an authorized administrator identity, which is represented by the runAsUser ID ADMIN-USERID.

- In the following example, either HIREN or NICK must approve the step:

```

<runAsUser>ADMIN-USERID</runAsUser>
<approver>HIREN NICK</approver>

```

- In the following example, both HIREN and NICK must approve:

```

<runAsUser>ADMIN-USERID</runAsUser>
<approver>HIREN</approver>
<approver>NICK</approver>

```

- In the following example, CONNER and either HIREN or NICK must approve:

```

<runAsUser>ADMIN-USERID</runAsUser>
<approver>CONNER</approver>
<approver>HIREN NICK</approver>

```

Using variable substitution

You can use variables to represent the runAsUser ID and approver ID. To indicate that substitution is used, include the attribute "substitution=true" on the element, and specify the substitution string on the element.

In the following example, the variable ADMIN-USERID is used to represent the runAsUser ID for a step:

```

<runAsUser substitution="true">${instance-ADMIN-USERID}</runAsUser>

```

If you use variable substitution, understand that the variable must be an instance variable; it cannot be a global variable.

In a called workflow, the caller scope variables are eligible for use with user ID substitution. For more information about caller scope variables, see [“Caller scope variables” on page 1169](#).

Static runAsUser IDs

If the runAsUser ID is resolved at workflow creation time, it is considered to be a *static* runAsUser ID. Here, the value is determined during substitution, by using values from the input properties file. Or, it might be set to a fixed value in the workflow definition.

At workflow creation time, z/OSMF checks the resulting value of substitution to ensure that it is a valid z/OS user ID and is permitted to z/OSMF (that is, permitted to the z/OSMF SAF profile prefix profile in the APPL class). If these checks fail, the workflow cannot be created.

As a recommended practice for a static runAsUser, include the variable in the workflow input properties file. Also, avoid using a variable that can be prompted for at workflow creation. For a description of the `prompt=` attribute for instance variables, see [“Using the element `atCreate` to qualify a variable definition” on page 1160](#).

Dynamic runAsUser IDs

If the runAsUser ID is assigned during workflow processing, it is considered to be a *dynamic* runAsUser ID. Suppose, for example, that Step A creates a user ID, which is then used by Step B for performing some action, such as issuing a command.

If a step is associated with a dynamic runAsUser ID, z/OSMF does not validate the runAsUser value at workflow creation time. Instead, the runAsUser value is checked when the step is run. When the step is being performed, the Workflows task processes the variable substitution to derive the actual user IDs for the step.

As a recommended practice for a dynamic runAsUser, use a variable that is not referenced in the workflow input properties file.

How the static or dynamic determination is made

The determination as to which runAsUser values are dynamic and which are static is made at workflow creation time, as follows:

- If a runAsUser value is defined with `substitution=true`, it is considered to be dynamic if the initial substitution results in no change to the value.
- If a runAsUser value is defined with `substitution=false`, or the initial substitution results in a change to the value, the runAsUser value is considered to be static.

Assume that a runAsUser value is represented by the variable reference `${instance-rau}`. [Table 582 on page 1153](#) shows how the static or dynamic determination is resolved for `${rau}`, depending on a number of factors, such as whether the variable can be prompted or is pre-specified in the input properties file.

Table 582. Basic usage scenarios: How static or dynamic determination is made for the runAsUser value						
<runAsUser>	<atCreate prompt=>	<runAsUser substitution=>	Input properties file?	Result of initial substitution	Dynamic or static?	Notes
<code>\${instance-rau}</code>	<i>false</i>	<i>true</i>	Not specified.	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.
<code>\${instance-rau}</code>	<i>true</i>	<i>true</i>	Not specified.	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.

Table 582. Basic usage scenarios: How static or dynamic determination is made for the runAsUser value (continued)						
<runAsUser>	<atCreate prompt=>	<runAsUser substitution=>	Input properties file?	Result of initial substitution	Dynamic or static?	Notes
<code>\${instance-rau}</code>	<i>false</i>	<i>true</i>	rau=IBMUSER	IBMUSER	Static	Static because the value changed after substitution.
<code>\${instance-rau}</code>	<i>true</i> ¹	<i>true</i>	rau=IBMUSER	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.

1. Workflows processing ignores the prompt=true setting for a runAsUser when it resolves the substitution string. Here, the runAsUser value is always determined to be dynamic.

Using translatable strings

The tables in “Workflow XML reference” on page 1173 indicate which elements have translatable values with a type of `nlsString` or `nlsRichString`.

A translatable string takes two optional attributes, `bundle=` and `bundleKey=`. If one attribute is specified, the other must also be specified.

Table 583 on page 1154 describes shows the `bundle=` and `bundleKey=` attributes.

Table 583. Translatable strings			
Attribute name	Description	Type	Requirements and restrictions
bundle=	The name of a bundle defined in the message manifest	A single-token string	The referenced bundle must exist.
bundleKey=	The key within a language file containing the replacement text for the text element.	<code>nonNullString</code>	The referenced file should contain a key of this name, but this is not validated by the Workflows task.

Using rich translatable strings

Within the schema, any string defined with type `nlsRichString` or `nlsRichVelocityString` is a translatable string that can contain HTML tags. Not all HTML tags (and their attributes) are supported, though tags for headings, tables, lists, hyperlinks, and text formatting are available.

The allowable tags are: `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `ol`, `ul`, `dl`, `dt`, `dd`, `li`, `br`, `p`, `hr`, `table`, `th`, `td` (with the `frame`, `rules`, and `width` attributes), `tr`, `caption`, `colgroup`, `col`, `thead`, `tbody`, `tfoot`, `i`, `b`, `u`, `em`, `strong`, `cite`, `code`, `samp`, `kbd`, `pre`, `tt`, `sub`, `sup`, `big`, `small`

To specify a hyperlink, use the anchor (`<a>`) tag. When clicked, the hyperlink opens a new tab or window, based on the user’s browser settings. You can specify the `href` attribute only. To include an ampersand character (`&`) in the URL, enclose the symbol in quotes: `"&"`. Also, include the protocol with the URL. For example, a value of `"http://www.ibm.com"` is correct, but `"www.ibm.com"` is not.

Defining variables for your workflow

This topic describes the elements and types that make up a variable definition. Variables can be referenced by workflow steps for substitution in step instructions and templates, and for calls to REST interfaces. A workflow can contain up to 1500 variable definitions.

This topic includes the following information:

- [“Using Velocity templates for variable substitution and other functions” on page 1155](#)
- [“Specifying the variable element and its attributes” on page 1156](#)
- [“Sub-elements of the variable element” on page 1158](#)
- [“Using the element atCreate to qualify a variable definition ” on page 1160](#)
- [“How to refer to a variable” on page 1161](#)
- [“Workflow internal variables ” on page 1163](#)
- [“System variables” on page 1165](#)
- [“Array variables” on page 1168](#)
- [“Caller scope variables” on page 1169](#)
- [“Providing a workflow variable input file” on page 1170](#)

The elements and attributes that are used to define variables are listed in [Table 596 on page 1214](#) and [Table 597 on page 1217](#).

Using Velocity templates for variable substitution and other functions

By using variables, you can add significant function to your workflow. With variables, you can design steps to prompt the user for input before performing particular functions, such as running a job. Further, you can use variables in conditional expressions, for added flexibility in your design.

In z/OSMF, the open source Apache Velocity Engine is used for performing variable substitution and conditional directives. You can use variables for simple string replacement, and also for creating conditional directives that allow you to generate different strings, based on the presence or value of any variable that is referenced by the step. The type of the variable, as defined in the XML, is passed in to the Velocity Engine so that the expected behavior is preserved, except for time and date, which are passed in as strings.

You define a variable by coding `$instance-variable-name`. The prefix *instance* means that the scope of the variable is within the current instance of the workflow. Another prefix, *global*, is used to define a variable that can be used by all workflows in the system.

The following example shows a workflow step that does not use variables.

```
<step name="Step1" >
  <title>
    Define the started task user ID to SAF.
  </title>
  <description>
    Define the started task user ID to SAF.
  </description>
  <instructions>
    You must define the user ID to your security product.
    For example, for RACF:<br/>
      ADDUSER STASK OMVS(UID(18136) HOME(/u/stask))
    <br/><br/>
    After you have entered this command from the TSO command line,
    press <strong>Finish</strong> to complete the step.
  </instructions>
  <weight>2</weight>
  <skills>Security administration</skills>
</step>
```

Suppose that you want to prompt the user for input, which might then be substituted in the command that is contained in the instructions. To do so, you can modify this step to include a variable. In the example that follows, the reference to UID 18136 in the previous example is replaced with a variable that is used to prompt the user for a UID.

```
<step name="Step1" >
  <title>
    Define the started task user ID to SAF.
  </title>
  <description>
    Define the started task user ID to SAF. You will be
    prompted for the UNIX UID to assign to the user.
  </description>
</step>
```

```

</description>
<variableValue name="uid" required="true"/>
<instructions substitution="true">
    You must define the user ID to your security product.
    For example, for RACF:<br/>
        ADDUSER STASK OMVS(UID($instance-uid) HOME(/u/stask))
    <br/><br/>
    After you have entered this command from the TSO command line,
    press <strong>Finish</strong> to complete the step.
</instructions>
<weight>2</weight>
<skills>Security administration</skills>
</step>

```

In the example, observe the following considerations:

- User input is defined by using the `variableValue` element. In this example, the variable is named `uid`.
- Substitution is performed by using a variable reference, which is `$instance-uid` in this example.

A variable reference follows this format:

- Dollar sign (\$)
- Scope, which is either `instance` or `global`
- Hyphen (-)
- Variable name, for example, `uid`.

For more examples of how to code symbolic variable references within instructions and templates, see file `workflow_sample_wizards.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

Note:

- When you are using Velocity comparison operators in the instructions, do not use the less than ("`<`") and greater than ("`>`") characters, as they interfere with XML. Instead, use the alternative notation: `lt`, `le`, `gt`, and `ge`.
- When you are doing string substitution, understand that the number sign ("`#`") is a reserved character for the Velocity Engine. Avoid using the number sign in string substitution, as it can lead to unexpected results when the variable is resolved.

If you need to use the number sign character in string substitution, use this technique:

1. Create a variable such as `numberSign` and assign it the value `"#"`
 2. Replace all of the `"#"` characters with the variable reference `"${instance-numberSign}"`.
- White space (newlines, tabs, spaces) is collapsed before text is displayed in the Workflows task. The indenting that is shown in this example is included for readability only. To obtain the required spacing for your workflow in the Workflows task, you must provide the appropriate HTML formatting tags. You might need to experiment with the spacing somewhat.

For more information about the Velocity Engine, see the following website: <http://velocity.apache.org>.

Specifying the variable element and its attributes

A workflow variable is defined on the `variable` (`<variable>`) element.

The elements and attributes that are used to define variables are listed in [Table 596 on page 1214](#) and [Table 597 on page 1217](#).

The following attributes are supported for the `variable` (`<variable>`) element:

name

Name of the variable. The variable name is required, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. The variable name must begin with a letter.

The combination of variable name and variable scope must be unique within the workflow.

When you are choosing a variable name, be aware that the following string values are reserved; they cannot be used as the variable name:

- workflowKey
- metaAppVersion
- releaseVersion
- String value that ends with --lastSetStep
- String value that ends with --lastSetTime.

scope

Scope of the variable, as follows:

instance

Variable is used only within the workflow in which it is defined. If multiple workflows are created from the same workflow definition file, each has its own set of instance scoped variables.

global

Variable can be referenced by any workflow that is imported into the Workflows task. Global variables are shared across all workflows, even workflows that are created from different workflow definitions. As an example, you might use a global variable to refer to a product-specific constant across a number of workflows that are associated with the product.

The scope is required. The default is *instance*.

The combination of name and scope must be unique within the workflow.

You cannot use the same name for both an instance variable and a global variable in the same workflow definition.

Use global variables with caution to avoid possible naming conflicts across unrelated workflows. Consider your naming conventions carefully and avoid using unspecific variable names. Similarly, consider qualifying your variables, for example, with the 3-character prefix associated with your software product, or a unique identifier.

Be aware that variables are case-sensitive. For example, "Variable1" is not the same as "variable1."

Notes:

- Global variables are deprecated, as of z/OS V2R3. IBM recommends that you use instance variables or system variables, instead. Global variables might not be supported in a future release. For information about creating system variables, see [“z/OS system variable services”](#) on page 802.
- z/OSMF includes a number of "built-in" workflow variables, which might provide the function that you require; see [“Workflow internal variables ”](#) on page 1163.

visibility

Specifies whether the variable is intended for public or private use. This attribute is intended for the workflow author's use. The visibility setting does not affect how the variable is processed by the Workflows task. This attribute is optional; the default is *private*.

Example of a variable definition

In the example in [Figure 499 on page 1158](#), the variable `variable_test` is defined.

```

<variable name="variable_test" scope="instance" visibility="private">
  <label>Variable 4</label>
  <abstract>Abstract for Variable 4.</abstract>
  <description>Description for Variable 4.</description>
  <category>variables</category>
  <string/>
</variable>

```

Figure 499. Specifying attributes on the variable element

Sub-elements of the variable element

This topic describes the sub-elements and types that make up a variable definition.

The `<variable>` element can contain the following sub-elements:

- label (required)
- abstract (required)
- description (required)
- exposeToUser (optional)
- category (required)
- datastore (optional)

Variables require a label (`<label>`) and an abstract (`<abstract>`). These values are displayed in the Workflows task when it prompts for input. In addition, the Workflows task displays a description (`<description>`) for the variable if the user clicks the information icon for the abstract.

You can specify a category (`<category>`) for a variable to assign it to a logical group of related variables. For a given step, all variables with the same category are displayed on the same web page. When viewed through the Workflows task, the workflows Step Perform wizard proceeds through each of the categories that you define for the step. In this way, you can logically organize many variables to provide context and an easier user experience for users who enter variable values.

Variable definition type-specific elements

A variable definition includes a type-specific element that contains elements and attributes specific to that type. The supported types and corresponding element names are described as follows:

boolean

The Workflows task displays a simple check box to prompt the user for this variable. Specify a default value of `true` or `false` to indicate whether the check box is initially displayed to the user as checked. If you do not provide a value, `true` is used by default.

string

The Workflows task displays the variable in a text box, initially primed with an optional default value, if you specify one. Use the `"multiline="` attribute (Boolean) of the string element (`<string>`) to specify whether the text box is small or large.

You can specify a number of choices for the variable. If so, the Workflows task displays the text box with a menu from which the user can select a value for the variable. The `"valueMustBeChoice="` attribute (Boolean) of the string element specifies whether the user must choose from the predefined values or can enter a custom value.

You can specify more validation criteria for the variable in one or more of the following ways:

- Minimum length (`<minLength>`) or maximum length (`<maxLength>`), or both, of the string value
- Predefined validation type (`<validationType>`) to be provided by the Workflows task. You can request validation for common constructs, such as data set names, data set qualifiers, z/OS user IDs. For a list of available validation types, see [“Variable definition elements and types summary” on page 1213.](#)

- Regular expression (<regularExpression>) that you provide when neither of the other mechanisms meet your requirements. The regular expression must adhere to the JavaScript standard; see the document posted at <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>.

The criteria that you specify are enforced by the Workflows task in real time as the user types the input value. If the user specifies incorrect syntax, for example, the Workflows task displays the variable in red, along with a default error message, which you can override by providing the error message element (<errorMessage>). Any default or choice value that you specify in the variable definition is also subject to these criteria.

The Workflows task cannot load an XML file that violates the criteria. Similarly, you cannot define the "valueMustBeChoice=" attribute with a value of `true` without defining any choices.

integer

The Workflows task displays the variable in a text box, initially set to a value that you can optionally specify. The Workflows task restricts the user's input value to a signed 31-bit value (in the range of -2147483648 to 2147483647). You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the integer. The Workflows task validates the default value against the minimum and maximum when the workflow definition file is imported into z/OSMF.

decimal

The Workflows task displays the variable in a text box, initially set to a value that you can optionally specify. A decimal is an integer with a "decimalPlaces=" attribute on the <decimal> element. A decimal value allows the same whole number value as an integer, plus up to six decimal places (that is, a value in the range of -2147483648.999999 to 2147483647.999999). The "decimalPlaces=" attribute has a default value of 1. You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the decimal. The Workflows task validates the default value against the minimum and maximum when the workflow definition file is imported into z/OSMF.

time

The Workflows task displays the variable in a timebox. By default, the time is displayed in 15-minute increments, based on your specified default, or the current time, if you do not specify a default. The user can type in a value, also. You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the time. The Workflows task restricts the user input to the range you specify.

Note:

- You specify this variable in hours-minutes-seconds format (hh:mm:ss), but the timebox displays the time in a slightly different format.
- The schema allows slight variations of this format, but the Workflows task does not. Using a time format other than hh:mm:ss can have an unpredictable result.

date

The Workflows task displays the variable in a calendar, for which the date is based on your specified value, or the current date, if you do not specify a value. The user can type in a value, also. You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the date. The Workflows task restricts the user input to the range you specify.

Note:

- You specify this variable in year-month-day format (yyyy-mm-dd), but the calendar displays the date in a slightly different format.
- The schema allows slight variations of this format, but the Workflows task does not. Using a date format other than yyyy-mm-dd might have an unpredictable result.

password

By defining a password variable, you can add a password prompt to your workflow. If you do so, the user is prompted to provide a password on the **Input Variables** tab of the Workflows task. In the user interface, the password variable is displayed as an input field. The input field replaces the user's typed characters with masking characters, such as asterisks ('*****'). The password is not shown as it is typed.

You must specify either of the following types of validation checking for a password variable:

- Minimum length (<minLength>) or maximum length (<maxLength>), or both, of the password value.
- Match with a regular expression (<regularExpression>). The expression must adhere to the JavaScript standard; see the document posted at <http://www.ecma-international.org/publications/files/ECMA-ST/ECma-262.pdf>.

The criteria that you specify are enforced by the Workflows task in real time as the user types the input value. If the user specifies an incorrect syntax, for example, the Workflows task displays the variable in red, along with a default error message, which you can override by providing the error message element (<errorMessage>).

The password variable type has no default value.

array

When you need to map a list of values or name-value pairs, use an array variable. The format of an array variable can be either a list of individual values (an *array list*) or a set of one or more name-value pairs (a *JSON array*).

Unlike other types of variables, an array variable cannot be set manually by the user from the Workflows task user interface (UI). Instead, an array variable must be set by using a workflow variable input file, or by using an output file in a workflow step (an inline template step or file template step).

An array variable has no default value.

For more information, see [“Array variables” on page 1168](#).

Using the element `atCreate` to qualify a variable definition

This topic describes the element `atCreate` (<atCreate>). For users of the Create Workflow REST service, the `atCreate` element provides additional options for working with variables.

The Create Workflow REST service is described in [“Create a workflow” on page 1023](#).

The following attributes can be specified on the `atCreate` element:

name

Specifies the variable for which the variable attributes are being set. The name is required. For example, to set a variable named `var1`, define the `atCreate` element with the name `var1`.

scope

Specifies the scope of the variable. This value is set to *instance* (the only valid value) or is omitted; the default is *instance*.

required

This attribute has a specialized purpose. For a workflow that is created through the Create Workflow REST service, this attribute indicates whether the variable must be set to a value at the time of workflow creation. A variable can be set in any of the following ways:

- Defining a default value for the variable in the workflow definition
- Setting a value for the variable in the workflow variable input file
- Specifying a value for the variable on a Create Workflow service.

This attribute is optional; the default is *false*. If a variable is marked as "required," but the variable is not given a value, an attempt to create the workflow through the Create Workflow REST service will fail with an error.

For a workflow that is created through the Workflows task user interface, this option is ignored. That is, the workflow is created, regardless of the setting of the required attribute.

Note: This setting is returned as the "requiredAtCreate" property of a variable by the Retrieve Workflow Definition service; See [“Retrieve a workflow definition” on page 1065](#).

prompt

For users of the Create Workflow REST service, this attribute identifies a variable that *should* be prompted for by the program that issues the REST service. By itself, the prompt attribute does not enforce any behavior for the workflow creation. However, by setting prompt to *true*, you can indicate that prompting is recommended for the variable. The user of the Create Workflow REST service can query the value of the prompt attribute for any variables in the workflow to determine whether any variables should be prompted for.

This attribute is optional; the default is *false*.

Note: This setting is returned as the "promptAtCreate" property of a variable by the Retrieve Workflow Definition service; See [“Retrieve a workflow definition”](#) on page 1065.

You can specify the atCreate element for any instance variables that are used in a workflow definition. The atCreate element is not valid for global variables.

Example of using the element atCreate

Suppose that you have a variable that would be useful to include in a number of different workflows. If so, you can define the variable in an XML file (an XML fragment) and include the same fragment in the appropriate workflow definitions. The variable is now shared between these workflows.

In [Figure 500 on page 1161](#), the variable `variable_test` is defined in an XML file. The variable is used by more than one workflow, so the variable definition is coded in an XML file that can be included with multiple workflows.

```
<variable name="variable_test" scope="instance">
  <label>Variable 4</label>
  <abstract>Abstract for variable 4.</abstract>
  <description>Description for variable 4.</description>
  <category>variables</category>
  <string/>
</variable>
```

Figure 500. Variable definition in this example

Now suppose that the variable's attributes for *required* and *prompt* need to be set differently for different workflows. The atCreate (<atCreate>) element, which is used with the variable element, allows you to specify different *prompt* and *required* settings for the same variable in different workflow definitions. To do so, have each workflow definition include the XML file that defines the variable. Then, in each workflow definition, specify the atCreate element to further clarify the properties of the variable.

In [Figure 501 on page 1161](#), the atCreate element is used to specify the attributes *required* and *prompt* for the variable `variable_test`, which was defined in [Figure 500 on page 1161](#). The atCreate element can be defined differently in any workflow definition that refers to the variable `variable_test`.

```
<atCreate name="variable_test" required="true" prompt="true"/>
```

Figure 501. How the atCreate element is used to specify variable attributes for required and prompt

By including the atCreate element in each workflow definition, you can set different values for the *prompt* and *required* attributes for the same variable in different workflows.

How to refer to a variable

Using braces around variable references is optional, but recommended as a good programming practice. The braces help to ensure that variables are clearly identified in the workflow. Further, the braces prevent ambiguity when it comes time for variable substitution, such as in conditional expressions, and jobs and scripts. For example, the variables `$st_userFRED` and `${st_user}FRED` are evaluated differently by

the Workflows task. In the former case, the Workflows task searches for a variable called `st_userFRED`. In the latter case, it is clear that the variable is `st_user`.

A variable reference can contain an underscore or a hyphen. For example, both of the following references are valid: `${instance_st_user} == "IBMUSER"` and `${instance-st_user} == "IBMUSER"`.

Examples of how variables are referenced and used are provided in [“Defining steps for your workflow”](#) on page 1117. Also, see the file `workflow_sample_variables.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

Simplified instance variable format in substitution and conditions

If you need to define many instance variables in a workflow definition file, you can save some typing by using the simplified variable format. Here, you can omit the prefix `instance-` from the names of instance variables. To use the simplified variable format, you must enable it by including the optional element `<workflowSettingInfo>`, which is a subelement of the `<workflow>` element.

The `<workflowSettingInfo>` element specifies variables settings for the workflow. It contains two subelements:

<variablesSetting>

To omit the prefix `instance-` from the names of instance variables, include the attribute `isInstanceVariableWithoutPrefix` set to `"true"`. By default, this attribute is `"false"`, which means that instance variables require the prefix.

If you set `isInstanceVariableWithoutPrefix` to `"true"`, you must also ensure that none of the instance variables in the workflow definition are prefixed by `instance-`, either in variable definitions or in conditional expressions.

<globalVariableGroup>

Specifies a global variable group name (on the `name=` attribute) for global variables in the workflow.

[Figure 502 on page 1162](#) shows how to specify the use of the simplified variable format.

```
<workflow>
<workflowSettingInfo>
  <variablesSetting isInstanceVariableWithoutPrefix="true"/>
  <globalVariableGroup name="GlobalVarGroup1" />
</workflowSettingInfo>
<workflowInfo>
...
```

Figure 502. Specifying the use of the simplified instance variable format in a workflow definition

Use care with the simplified variable format when you specify variables in substitutions to ensure that the variables are specified consistently. In the following example, the simplified format is used for instance variable references. In the example, Step 3 is not performed unless Step 1 and Step 2 complete with a return code zero and the instance variable `${st_user}` is `IBMUSER`.


```

<Step name=Step3">
  <title>A conditional step based on return code</title>
  <description>This conditional step is not ready unless
                the two preceding steps complete with RC 0
                variable st_user value is IBMUSER
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression><![CDATA[${step1.returnCode} == "0000" &&
                        ${step2.returnCode} == "0000" &&
                        ${st_user} == "IBMUSER" ]]>
    </expression>
    <description>This conditional step is not ready unless the two
                preceding steps complete with RC 0 and the variable
                st_user value is IBMUSER.</description>
  </condition>
  <targetState>Ready</targetState>
  :
</template>
<inlineTemplate substitution="true">
//STEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
ADDGROUP ${st_group} OMVS(GID(${st_gid}))
/*
</inlineTemplate>
<submitAs>JCL</submitAs>
</template>
</step>

```

Figure 503. You can use variable values in the condition to be satisfied.

Note: Though the simplified variable format is supported, the suggested practice is that you use the standard format, for example: "**\${instance-varName}**" or "**\${global-varName}**".

Workflow internal variables

z/OSMF includes a number of "built-in" variables, which can be used by workflow authors. These variables are called *workflow internal variables*. You can reference them in your workflow definition without the need for you to define them. This topic lists the workflow internal variables that are available for your use.

Workflow internal variables are separated into two different scopes, as follows:

Step scope

Internal variables that refer to information about the specific step in which they are referenced. For example, the step title. To refer to these variables in your workflow, use the following syntax: **\$_step-variable-name**

Workflow scope

Internal variables that refer to information about the entire workflow. For example, the workflow name. To refer to these variables in your workflow, use the following syntax: **\$_workflow-variable-name**

A reference to a workflow internal variable must include the scope prefix: **_step-** or **_workflow-**. Otherwise, the variable is treated as an undeclared local variable, which results in an error.

Table 584 on page 1164 lists the variables that are provided with z/OSMF. Each variable resolves to a string, which is described in the **Description** column. Some internal variables are designed specifically for use with provisioning workflows. Others are applicable to any workflow type.

Table 584. Internal variables: Variables that are provided with z/OSMF.

Variable reference syntax	Variable scope	For use with (workflow type)	Description
<code>\${_workflow-actionID}</code>	Workflow	Provisioning workflows	For an actions workflow, this variable resolves to the action ID for the action object.
<code>\${_workflow-clusterInstanceName}</code>	Workflow	Provisioning workflows	Resolves to the created cluster instance name.
<code>\${_workflow-domainID}</code>	Workflow	Provisioning workflows	Resolves to the ID of the domain that is associated with the template.
<code>\${_workflow-parentRegistryID}</code>	Workflow	Provisioning workflows	Resolves to the ID of the software instance parent registry entry.
<code>\${_workflow-registryID}</code>	Workflow	Provisioning workflows	Resolves to the ID of the software services registry.
<code>\${_workflow-softwareServiceInstanceName}</code>	Workflow	Provisioning workflows	Resolves to the created software service instance name.
<code>\${_workflow-sysplexName}</code>	Workflow	All workflow types	Resolves to the name of the sysplex on which the workflow is running.
<code>\${_workflow-systemName}</code>	Workflow	All workflow types	Resolves to the name of the system on which the workflow is running.
<code>\${_workflow-tenantID}</code>	Workflow	Provisioning workflows	Resolves to the ID of the tenant that is associated with the resource pool.
<code>\${_workflow-templateID}</code>	Workflow	Provisioning workflows	Resolves to the unique identifier for the template.
<code>\${_workflow-templateName}</code>	Workflow	Provisioning workflows	Resolves to the name of the template that is associated with the resource pool.
<code>\${_workflow-workflowName}</code>	Workflow	All workflow types	Resolves to the descriptive name for the workflow.
<code>\${_workflow-workflowKey}</code>	Workflow	All workflow types	Resolves to the workflow key, which is a string value, generated by z/OSMF to uniquely identify the workflow instance.
<code>\${_workflow-workflowOwner}</code>	Workflow	All workflow types	Resolves to the user ID of the workflow owner.

Table 584. Internal variables: Variables that are provided with z/OSMF. (continued)

Variable reference syntax	Variable scope	For use with (workflow type)	Description
<code>\${_workflow-workflowOwnerUpper}</code>	Workflow	All workflow types	Resolves to the user ID of the workflow owner (in uppercase letters).
<code>\${_workflow-workflowSystem}</code>	Workflow	All workflow types	Resolves to the unique name that is assigned to the system definition (the system nick name).
<code>\${_step-runAsUser}</code>	Step	All workflow types	Resolves to the user ID under which the workflow step is to be performed (a runAsUser ID).
<code>\${_step-runAsUserUpper}</code>	Step	All workflow types	Resolves to the runAsUser ID under which the workflow step is to be performed (in uppercase letters).
<code>\${_step-stepName}</code>	Step	All workflow types	Resolves to the step name for the current step.
<code>\${_step-stepNumber}</code>	Step	All workflow types	Resolves to the step number for the current step. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.
<code>\${_step-stepOwner}</code>	Step	All workflow types	Resolves to the user ID of the user to whom the step is assigned.
<code>\${_step-stepOwnerUpper}</code>	Step	All workflow types	Resolves to the user ID of the user to whom the step is assigned (in uppercase letters).
<code>\${_step-stepTitle}</code>	Step	All workflow types	Resolves to the step title for the current step.

System variables

It is possible to create variables that you can use with all workflow instances. These variables are called *system variables*. Unlike instance variables, system variables are system-wide in scope. You can reference them in your workflow without the need for you to define them in the workflow definition. This topic explains how to create and use system variables.

To create a system variable, you use z/OSMF system variable services, or the z/OSMF Systems task GUI. With these functions, you can create, update, retrieve, and delete z/OSMF system variables. The system variables that you create are stored by z/OSMF in the z/OS system variable pool. Each z/OS system in your enterprise can have a unique set of system variables in its system variables pool.

You can create any number of system variables on a z/OS system. No practical limit exists.

A system variable cannot be used as part of an array variable. For information about array variables, see [“Array variables” on page 1168](#).

How to include system variables in your workflow

The following topics describe ways of including system variables in your workflow:

- “Using direct variable reference” on page 1166 shows the use of the `setVariable` function to map a system variable to a workflow variable.
- “Using z/OSMF system variable services” on page 1166 shows the use of REST services to drive the creation of a workflow.

Using direct variable reference

The following steps describe a technique for including a system variable in a workflow.

1. **Create the system variable.** If a desired system variable is not already defined on your system, you can create it. Suppose, for example, that you want to define a variable for JES job class so that you can specify the job class for workflow-initiated jobs. In z/OSMF, open the Systems task. In the **Systems** table, select your system. Then, select **Actions > System Variables** to display a window that you can use to create the job class system variable or any other system variables that you require.
2. **Map the system variable to a variable in your workflow.** In your workflow, include a step that assigns the value of the system variable to a workflow variable. To do so, you can use the `setVariable` function.

In the following example, the `setVariable` function is used to assign the value of the system variable you defined (JOBCLASS) to the workflow variable `wf_JOBCLASS`.

```
<step name="AssignSysVariable" optional="false">
<title>AssignSystemVariable</title>
<description>Assign system variable to workflow variable</description>
<instructions substitution="false">Generated instruction text for this step.
Update this field with your own text</instructions>
<weight>1</weight>
<autoEnable>true</autoEnable>
<canMarkAsFailed>false</canMarkAsFailed>
<setVariable name="wf_JOBCLASS" scope="instance" substitution="true">${_sys-JOBCLASS}</setVariable>
</step>
```

Figure 504. Example of how a system variable can be used in a workflow step.

Notice that the format for referring to a system variable is `${_sys-variableName}`. This notation is used to extract the system variable value from the system variable pool. You can include the system variable directly in the workflow by using the `${_sys-variableName}` format, which is similar to how you use `${instance-varName}` to refer to an instance variable defined in the workflow. Depending on the system where workflow runs, the system variable that is associated with that system is used.

Using z/OSMF system variable services

In contrast to the previous technique, the following method makes use of z/OSMF REST services to accomplish the same goal.

1. **Create system variable.** From a program that you create, issue the "Create or update system variables" REST service to define the system variable to your system. For more information, see “Create or update system variables” on page 803.
2. **Get the system variable.** From your program, issue the "Get system variables" service to obtain the system variables for the desired system. This operation retrieves the system variables from the system variable pool and returns them in a JSON array. Your program should include logic to examine the array for the system variables of interest. For more information, see “Get system variables” on page 805.

In the example in [Figure 505 on page 1167](#), the returned array includes a system variable for the JES job class.

```
{ "system-variable-list":
  [
    { "description": "Class",
      "name": "JOBCLASS",
      "value": "X" }
  ]
}
```

Figure 505. Example of the returned object from the Get system variables service.

3. **Map the system variable to a variable in your workflow.** Your program can create the workflow by issuing the "Create workflow" service. In the request object for this service, assign the value of the system variable to the variable array that is passed to the workflow on creation. For more information, see ["Create a workflow"](#) on page 1023.

Example use for a system variable

Figure 506 on page 1167 shows how a system variable can be used in the z/OSMF user interface. Here, the system variable JOBCLASS is substituted for CLASS in the customized JOB statement for z/OSMF.

Workflows > Customize JOB Statements

Customize JOB Statements

Selected JOB statement:
Default JOB Statement

* JOB statement JCL:

```
//IZUWFJB JOB (ACCTINFO),CLASS=${_sys-JOBCLASS},MSGCLASS=0,
//MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
```

Undo Changes

Figure 506. Variable substitution for a system variable in the Workflows task customize job statements window.

The resulting JCL substitutes the system variable with CLASS=A when the workflow runs on one system and CLASS=X when it runs on a different system. See [Figure 507 on page 1167](#).

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
//MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
/*JOBPARM SYSAFF=Z2
//ALLOC1 EXEC PGM=IEFBR14
//DD1 DD DSN=HIREN.SLC.DEMO,
//DISP=(NEW,CATLG),
//UNIT=3390,SPACE=(TRK,(10,10,10)),
//DCB=(DSORG=PO,RECFM=FB,LRECL=1024)
```

Rectangular Snip

```
//IZUWFJB JOB (ACCTINFO),CLASS=X,MSGCLASS=0,
//MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
/*JOBPARM SYSAFF=Z3
//ALLOC1 EXEC PGM=IEFBR14
//DD1 DD DSN=HIREN.SLC.DEMO,
//DISP=(NEW,CATLG),
//UNIT=3390,SPACE=(TRK,(10,10,10)),
//DCB=(DSORG=PO,RECFM=FB,LRECL=1024)
```

Figure 507. Results of variable substitution for a system variable in the Workflows task.

Array variables

When you need to map a list of values or name-value pairs, use an array variable. Array variables provide an alternative to defining multiple variables to represent multiple values.

The format of an array variable can be either a list of individual values (an *array list*) or a set of one or more name-value pairs (a *JSON array*).

Unlike other types of variables, an array variable cannot be modified manually by the user from the Workflows task user interface (UI). Instead, an array variable can be set by using an output file in a workflow step (an inline template step or file template step). Therefore, the Workflows task does not display array variables to the user.

Notes:

1. An array variable has no default value.
2. A system variable cannot be used as part of an array variable. For information about system variables, see [“System variables” on page 1165](#).
3. An excessively large properties file with many thousands of array variables can degrade the performance of the Workflows task.

Examples of using array variables

Examples:

- This array variable is formatted in a list of individual items:

```
["Z0SV24T", "DB211T"]
```

- This array variable is formatted as a JSON array (name-value pairs):

```
[{"property1": "tt1", "dsName": "TEST.DSNAME.TT1"},  
 {"property1": "tt2", "dsName": "TEST.DSNAME.TT2"},  
 {"property1": "tt3", "dsName": "TEST.DSNAME.TT2"}]
```

- This array variable combines both list and JSON array formats:

```
["Z0SV24T", "DB211T",  
 {"property1": "tt1", "dsName": "TEST.DSNAME.TT1"}]
```

Performing substitution with array variables

As with other types of variables, substitution with array variables follows the conventions of the velocity template. For more information, see [“Using Velocity templates for variable substitution and other functions” on page 1155](#).

In the following example, the array variable `testJsonArrayVariable` is used in a script that is run by a workflow step.

```
#foreach($test in $instance-testJsonArrayVariable)  
#set($test.dsn=$test.dsn + 500)  
$test.dsn  
#end
```

Figure 508. Example of how an array variable might be used in a workflow step.

The array variable is treated as a string during substitution in the workflow, as shown in [Figure 509 on page 1169](#).

```

<step name="complexCondition"
<title>Install Product ABC</title>
<description>In this step, both prereqTargetStateSet and normal targetStateSet are used together.</
description>.
<prereqStep name="basicStep1"/>
<prereqDescription>If Basic Step 1 is completed, this step becomes ready to perform.</prereqDescription>
<prereqTargetStateSet>
<condition>
  <expression><![CDATA[${instance_testJsonArrayVariable} ==
'[{ "dsn": "TEST.SYS1.LINKLIB", "dstype": "PDS" }]']>
  </expression>
<description>Example of using an array variable</description>
<!-- Target step state (skipped) is specified in this conditional step -->
<targetState>Skipped</targetState>
</condition>

```

Figure 509. Example of how an array variable might be used in a workflow step.

In the example in [Figure 509 on page 1169](#), if the value of the variable `$instance_testJsonArrayVariable` is `[{"dsn": "TEST.SYS1.LINKLIB", "dstype": "PDS"}]`, the step is skipped.

Caller scope variables

When a workflow calls another workflow, the calling workflow's instance variables are implicitly shared with the called workflow. Such variables are known as *caller scope* variables.

A called workflow can reference its caller scope variables by using the following syntax:

```
${_caller-VAR1}
```

This method provides a simple alternative to explicitly mapping the caller's variables in the called workflow, as described in [“Sharing variables between the calling workflow and called workflow” on page 1140](#).

To be shared with a called workflow, the instance variables must be visible for public use. On the variable element (`<variable>`), the visibility attribute specifies whether the variable is intended for public or private use. This attribute is optional; the default is `private`. To allow an instance variable to be shared with a called workflow, ensure that the visibility attribute is set to `public`. For example:

```
<variable name="VAR1" scope="instance" visibility="public">
```

Caller scope variables are:

- Public visibility instance variables only.
- Shared only with the called workflow.
- Available only to **scope=none** called workflows. A `scope=none` workflow is created as a new instance whenever it is called.
- Static. A copy of the caller variables is made from the calling workflow when the called workflow is created. These variables are not updated to reflect the variable state in the calling workflow after the called workflow is created.
- Read only. If the called workflow must write to a variable in the calling workflow, use workflow-to-workflow variable mappings instead.
- Eligible for use in substitution. For example:

```
/u/userid/${_caller-FOLDER}/${_caller-FILE}
```

Information about caller scope variables is not returned to callers of the Get Workflow Properties REST API. For information about this API, see [“Get the properties of a workflow” on page 1031](#).

Providing a workflow variable input file

This topic describes the format of the editable properties file that is called the *workflow variable input file*. With this file, you can supply preset values for the variables that you use in your workflow definition file. By including a variable input file with your workflow definition file, you save users from having to manually enter values for some or all of the variables in your workflow.

How a workflow variable input file is used

A workflow variable input file is an optional properties file that you, the workflow author, can use to pre-specify one or more of the input variables that are defined in the workflow definition. By supplying variable values in this way, you make it possible for the user to create a workflow without having to interactively enter the inputs in the Workflows task Step Perform wizard. The Workflows task treats any variables set through the workflow variable input file the same as if the user entered them manually.

If you provide a workflow variable input file, include it with the other materials that you supply to the workflow user, such as the workflow definition file and the other files that comprise your workflow definition. Ensure that the documentation for your workflow definition makes note of the file name, and provides any related instructions for editing or storing the file on the user's z/OS system. The Workflows task accesses the workflow variable input file under the user's identity, thus, the file must be read-accessible by the user who is creating the workflow.

At workflow creation time, the user imports the workflow variable input file into the Workflows task, along with the workflow definition file. The Workflows task reads in the contents of the file and saves its values for use with the created workflow. The Workflows task uses the variable input file in addition to any global variables that are already defined to Workflows task. Any new variables that are defined with a global scope become available to the other workflow instances on the user's system. After the Workflows task imports the file, the task no longer refers to the file.

Creating a workflow variable input file

As the workflow author, you can create a workflow variable input file as a text file, by using an editor of your choice. Use a file type of `.txt` or `.properties`. The following encoding formats are valid:

- ASCII or IBM-1047 (EBCDIC) for workflow variable input files in UNIX files.
- IBM-1047 for workflow variable input files in z/OS data sets.

In the variable input file, specify the properties (variables and their respective values) as one or more key-value pairs. Valid separator characters are equal signs (=), colons (:), or spaces. [Figure 510 on page 1170](#) shows the valid formats for specifying properties in the variable input file.

```
key1 = value1
key2 : value2
key3  value3
```

Figure 510. Format of a workflow variable input file

[Figure 511 on page 1170](#) shows an example of the contents of a workflow variable input file.

```
Boolean1 = false
String3  = SYS1.LINKLIB
Integer2  35
Decimal2  : 3.3
Time2     03:03:00
Date1     = 2013-11-11
```

Figure 511. Example of a workflow variable input file

The example in [Figure 511 on page 1170](#) is designed to work with the file `workflow_sample_variables.xml`, which is supplied as a sample with z/OSMF. For a description of this file and other coding examples, see [“Sample XML files for your reference” on page 1105](#).

The Workflows task does no syntax checking of the properties that are specified in the workflow variable input file. Therefore, you must ensure that valid values are specified for each of the properties.

Also, observe the following considerations:

- Each property that is specified in the variable input file must correspond to a variable named in the workflow definition file. Otherwise, the Workflows task ignores the property.
- If the variable input file specifies a property that matches a variable that is already defined to the Workflows task as a global variable, the Workflows task detects the conflicting definitions and prompts the user for a selection. See [“Avoid conflicting variable definitions” on page 1172](#)

You can provide the workflow variable input file in either a z/OS UNIX file or a z/OS data set. For a z/OS data set, use a sequential data set, a member of a partitioned data set (PDS), or partitioned data set extended (PDSE).

If you create the workflow variable input file on a workstation, it is recommended that you use File Transfer Protocol (FTP) in binary mode to transfer the XML files to a z/OS system. Doing so helps to ensure that the files are encoded properly for use on z/OS.

Using variable substitution in the workflow variable input file

It is possible to use variable substitution in the workflow variable input file. With this ability, you can define a value once in the file and refer to that value on subsequent lines to use the same value.

To use variable substitution in the workflow variable input file, you must include the following specification in the file:

```
_IZU_VARIABLE_SUBSTITUTION_ON
```

Position this specification before any lines that refer to the variable.

Similarly, you can turn off variable substitution for subsequent lines in the file by including the following specification in the workflow variable input file:

```
_IZU_VARIABLE_SUBSTITUTION_OFF
```

Thereafter, subsequent variables are interpreted as literal values.

If you omit `_IZU_VARIABLE_SUBSTITUTION_ON` from workflow variable input file, the default is that no variable substitution is performed for the file.

[Figure 510 on page 1170](#) shows an example of how you can turn on and turn off variable substitution in the same workflow variable input file.

```
base = HelloWorld
_IZU_VARIABLE_SUBSTITUTION_ON
var = ${base}
_IZU_VARIABLE_SUBSTITUTION_OFF
var2 = ${base}
```

Figure 512. Using variable substitution in a workflow variable input file

In [Figure 510 on page 1170](#), notice that variable substitution is:

- On when the variable `var` is processed. As a result, the variable is replaced by the string `HelloWorld`.
- Off when the variable `var2` is processed. As a result, the variable is replaced by the string `${base}`.

Figure 511 on page 1170 shows how the previous example would be processed in the Workflows task Step Perform wizard. In this example, a variable input file that is named `VariableInput.properties` is provided for the workflow. Notice that substitution is performed for the first variable `var`, but not for the second variable `var2`.

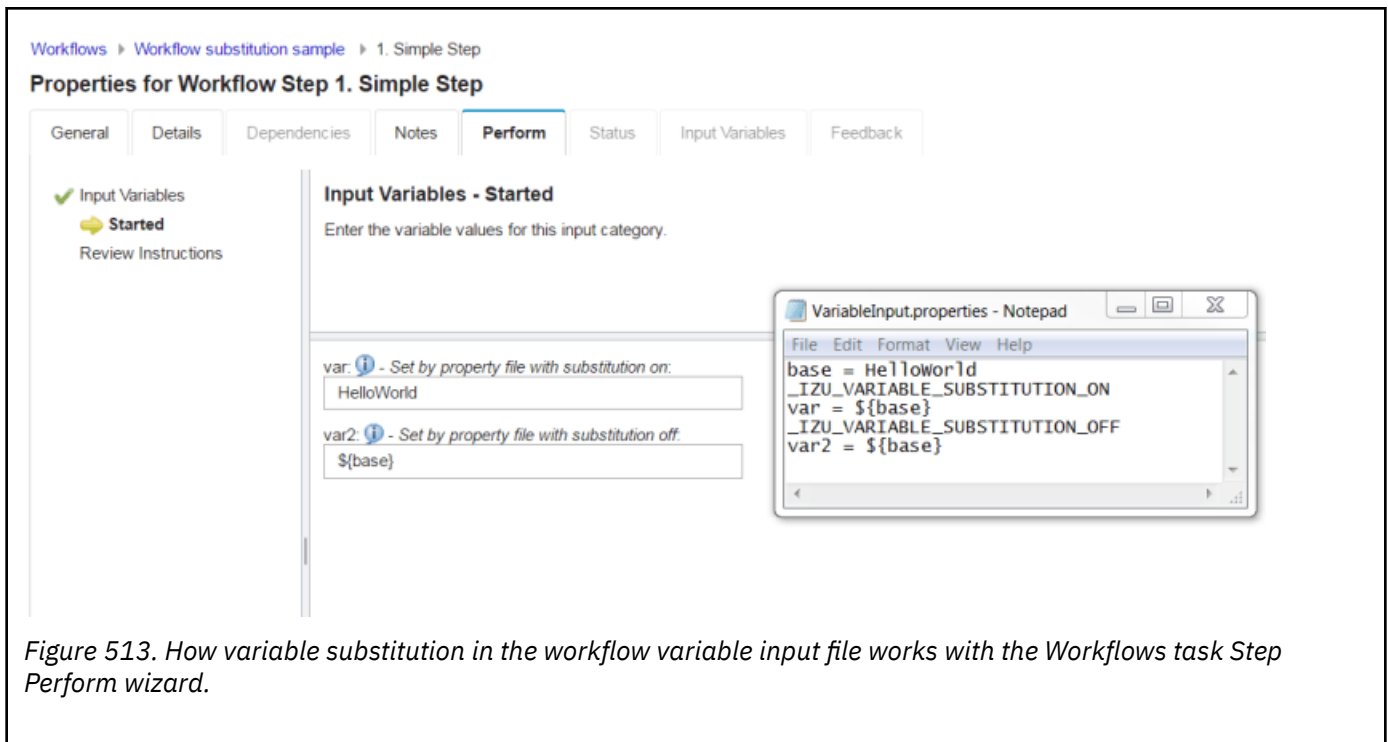


Figure 513. How variable substitution in the workflow variable input file works with the Workflows task Step Perform wizard.

Avoid conflicting variable definitions

If the Workflows task detects that an imported variable conflicts with an existing global variable, the user is prompted to choose the appropriate value. The user can select to use the input file variable in place of existing global variables, or ignore the input file variable, and use the existing global variable instead. The user's selection determines which version of the variable is saved in the Workflows task global variable pool for use with other workflow instances. Thus, the user's selection affects any other workflows that refer to the same global variable.

It is recommended that you choose unique names for variables to avoid possible naming conflicts with unrelated workflows. Consider your naming conventions carefully and avoid using unspecific variable names. Similarly, consider qualifying your variables, for example, with the three-character prefix associated with your software product, or a unique identifier.

Depending on your design, you might determine that the output file variables must always be used in place of a workflow's existing instance variables. If so, you can include the `needResolveConflicts` attribute (needResolveConflicts) on the output subelement, and set it to false. If so, the Workflows task uses the output file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden. The default is true; if variable conflicts exist, the user is prompted to resolve the conflicts.

For an array variable, you can use the `load-output-file-array-value` attribute (`loadOutputFileArrayValue`) to manage potential variable conflicts. If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task. Otherwise, if this attribute is set to false, the workflow uses the existing values from the Workflows task.

Workflow XML reference

This language reference describes the elements and attributes that comprise a workflow definition.

This topic is organized in tables, with each table describing a major portion of the Workflows schema—the elements and their attributes, default values, the XML attribute data types, and whether a particular attribute is required. [Table 585 on page 1173](#) lists the reference tables.

Table 585. Reference tables for the Workflows XML schema		
Element type	Description	Where described
Workflow metadata elements	The elements that make up the workflow metadata	Table 586 on page 1175
Sub-elements for a configuration type workflow	For a configuration type workflow, these sub-elements are required	Table 587 on page 1179
Sub-elements for a provisioning type workflow	For a provisioning type workflow, these sub-elements are required	Table 588 on page 1181
Workflow upgrade elements	The elements that define the workflow upgrade options	Table 589 on page 1183
Manifest elements	The elements that make up the manifest	Table 590 on page 1185
Step elements: Elements to use for defining all steps	The elements to use for defining all steps	Table 591 on page 1187
Step elements: Additional sub-element for parent steps	The step sub-element, which is used to identify the containing step as a parent step	Table 592 on page 1189
Step elements: Additional sub-elements for leaf steps	The sub-elements to use for defining a leaf step, which is a step that does not contain step elements	Table 593 on page 1191
Step elements: Additional sub-elements for REST steps	The sub-elements to use for defining a step that issues a REST request, such as GET or PUT. This type of step is referred to as a <i>REST step</i> .	Table 594 on page 1206
Step elements: Additional sub-elements for steps that invoke another workflow	The sub-elements to use for defining a step that invokes another workflow, which is referred to as the <i>called workflow</i> .	Table 595 on page 1210
Variable definition elements	The elements that make up a variable definition.	Table 596 on page 1214
Variable definition type-specific sub-elements	The type-specific sub-elements that make up a variable definition.	Table 597 on page 1217
atCreate element	For users of the Create Workflow REST service, the atCreate element provides additional options for working with variables.	Table 598 on page 1221

In the tables that follow, the elements are listed in the order in which they are required by the schema. Though you can omit optional elements, the elements that you specify must follow the order in which the elements are presented. In contrast, the attributes within an element can be specified in any order.

Note: The tables in this language reference are formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Workflow metadata elements summary

Table 586. Workflow metadata elements

Element name	Description	Required or optional	Type	Supported attributes
workflowSettingInfo	Specifies variables settings for the workflow.	Optional.	Contains the following sub-elements: <ul style="list-style-type: none"> variablesSetting (optional) globalVariableGroup (optional) 	<p>The following attribute is supported for the <i>variablesSetting</i> element:</p> <p>isInstanceVariableWithoutPrefix Indicates whether the simplified format is used for variable name references. If set to "true," the prefix instance- must be omitted from references to instance variables, such as in variable definitions and in conditional expressions.</p> <p>The following attribute is supported for the <i>globalVariableGroup</i> element:</p> <p>name Specifies the global variable group name for global variables in the workflow.</p>
workflowInfo	Contains the workflow metadata.	Required.	Contains the following sub-elements: <ul style="list-style-type: none"> parallelSteps (optional) workflowID (required) workflowDescription (required) workflowVersion (required) vendor (required) configuration or general (optional) 	No attributes are supported for this element.
parallelSteps	For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, this property is yes. Otherwise, if this property is omitted or set to no, automated steps are run one-by-one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.	Optional	Boolean	No attributes are supported for this element.

Table 586. Workflow metadata elements (continued)

Element name	Description	Required or optional	Type	Supported attributes
workflowID	A short, arbitrary value that identifies the workflow.	Required.	nonNullString	<p>The following attributes are supported for the <i>workflowID</i> element:</p> <p>scope Indicates the singleton scope for the workflow. The following values are valid:</p> <p>system A maximum of one instance of this workflow can exist on any one system in the sysplex.</p> <p>sysplex A maximum of one instance of this workflow can exist in the sysplex.</p> <p>none No limit exists for the number of instances of this workflow. For a callable workflow, this setting means that a new instance is always created on the calling system.</p> <p>The default setting is <i>none</i>. Omitting the scope attribute has the same effect as the default setting.</p> <p>isCallable Indicates whether the workflow can be called by another workflow, and, if so, the callable range for the workflow. The following values are valid:</p> <p>system This workflow can be called only by another workflow that is running on the same system.</p> <p>sysplex This workflow can be called by any workflow that is running in the same sysplex.</p>
workflowDefaultName	Default name for the workflow. This value is shown in the Workflow name field of the Workflows task when a user creates the workflow. If you omit this value, the Workflows task creates a name for the workflow.	Optional.	nonNullString	No attributes are supported for this element.
workflowDescription	A short description of the workflow.	Required.	nlsString	No attributes are supported for this element.

Table 586. Workflow metadata elements (continued)				
Element name	Description	Required or optional	Type	Supported attributes
workflowVersion	The version of this workflow definition file. Update this value whenever you change any portion of the workflow definition file, including changes to the primary XML file or any or referenced files. The Workflows task caches only the latest version of any imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you author a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.	Required.	nonNullString	No attributes are supported for this element.
vendor	The name of the workflow provider.	Required.	nonNullString	No attributes are supported for this element.
<i>Workflow category</i> is a classification of the activities to be performed in the workflow. To indicate a category, specify one of the following elements: <i>configuration</i> , <i>provisioning</i> , or <i>general</i> . Specifying a workflow category element is optional. By default, the workflow category is <i>general</i> . The category elements are described in the next rows.				
Configuration	A workflow that is used to configure system software is classified as a <i>configuration workflow</i> .	Optional.	Contains the sub-elements that are listed in Table 587 on page 1179 .	No attributes are supported for this element.
Provisioning	A workflow that is used to provision system software is classified as a <i>provisioning workflow</i> .	Optional.	Contains the sub-elements that are listed in Table 588 on page 1181 .	No attributes are supported for this element.
General	All other workflows are classified as <i>general workflows</i> .	Optional.	Empty	No attributes are supported for this element.

Table 587. Sub-elements for a configuration type workflow			
Element name	Description	Required or optional	Type
productID	A short, arbitrary value that identifies the workflow.	Required.	nonNullString
productName	The name of the product	Required.	nonNullString
productVersion	The product version	Required.	nonNullString

Table 588. Sub-elements for a provisioning type workflow			
Element name	Description	Required or optional	Type
productID	Identifier of the product or component that is being provisioned by the workflow, such as the product identifier (PID) or function modification identifier (FMID).	Required.	nonNullString
productName	Name of the product or component that is being provisioned by the workflow.	Required.	nonNullString
productVersion	Version and release of the product or component that is provisioned by the workflow.	Required.	nonNullString
softwareType	Type of software to be provisioned by the workflow.	Required.	

Workflow upgrade elements summary

Table 589. Workflow upgrade elements			
Element name	Description	Required or optional	Type
preserveOptions	Contains the workflow upgrade options.	Optional.	Contains the following sub-elements: <ul style="list-style-type: none"> • version (required) • variableSet (optional) • stepSet (optional) • workflowHistory (optional) • workflowNotes (optional) • include (required) • exclude (optional) • upgradeNotes (optional)
version	Identifies the workflow version that can be upgraded by this workflow definition file.	Required; at least one.	Contains the following sub-elements: <ul style="list-style-type: none"> • value (required) • type (required)
variableSet	The variables to copy to new workflow instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> • defaultChecked (optional)
stepSet	The steps to copy to new workflow instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> • defaultChecked (optional)
workflowHistory	Specifies whether to copy the workflow history from the existing workflow to the new instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> • defaultChecked (optional)
workflowNotes	Specifies whether to copy the workflow notes from the existing workflow to the new instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> • defaultChecked (optional)
include	Specifies the step or variable defined by prior workflow definition file to be copied. Can be specified multiple times. It supports regular expression or variable name.	Required.	Contains the following sub-elements: <ul style="list-style-type: none"> • name (optional) • mapTo (optional) • regExp (optional)
exclude	Specifies the variables to exclude from the set that is generated by <include> elements.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> • name (required)

Manifest elements summary

Table 590. Manifest elements summary table

Element name	Description	Required or optional	Type	Supported attributes
translatedTextFiles	Contains the language file definitions.	Optional.	If specified, it must contain the following sub-elements: <ul style="list-style-type: none"> • bundle (required) • language (required) 	No attributes are supported for this element.
bundle	Contains the set of language files for the bundle	Required if the translatedTextFiles element is specified. A workflow can contain 1 — 500 bundles.	A sequence of language elements.	The following attribute is supported for the <i>bundle</i> sub-element: <p>name The name of the bundle. The name is required, and must be a single-token string.</p>
language	Locates a file for a particular language.	Required if the bundle element is specified. A bundle can contain 1 — 10 languages.	Empty	<p>The following attributes are supported for the <i>language</i> sub-element:</p> <p>name The language identifier as defined in the XML standard and RFC 3066, but using only the language portion without the country suffix. The identifier is required, and must be unique within a bundle.</p> <p>path The path name of the language file. The path is required, and its data type must be a nonNullString. A language file can be a UNIX file, a sequential data set, or a PDS member. The path name format is described in “References to external files” on page 1106.</p>

Step elements summary

Table 591. Step elements summary table: Elements to use for defining all steps

Element name	Description	Required or optional	Type	Supported attributes
step	Contains attributes of a step. Up to 500 step elements can be defined in a workflow.	Required. At least one step must be defined.	<p>The step element can contain the following sub-elements:</p> <ul style="list-style-type: none"> • title (required) • description (required) • prereqStep (optional) <ul style="list-style-type: none"> • If a step is a parent step (contains step elements), use the sub-element that is listed in Table 592 on page 1189 to define the step. • If a step is a leaf step (contains no step elements), use one or more of the sub-elements that are listed in Table 593 on page 1191 to define the step. • If a step invokes another workflow, use one or more of the sub-elements that are listed in Table 595 on page 1210 to define the step. 	<p>The following attributes are supported for the <i>step</i> element:</p> <p>name The name of the step. The name is required, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter, and must be unique within the workflow.</p> <p>optional Indicates whether the step is optional. This attribute is optional. If specified, its data type must be Boolean. By default, the value is <i>false</i>.</p>
title	A short description of the task.	Required.	nlsString	No attributes are supported for this sub-element.
description	A more detailed description of the step.	Required.	nlsRichString	No attributes are supported for this sub-element.
prereqStep	Identifies a step that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.	Optional.	Empty	<p>The following attribute is supported for the <i>prereqStep</i> sub-element:</p> <p>name Name of the prerequisite step. The name is required, must refer to a defined step, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must start with a letter.</p>

Table 592. Step elements summary table: Additional sub-element for parent steps

Element name	Description	Required or optional	Type	Supported attributes
step	<p>A substep. Existence of this sub-element defines a parent step. This sub-element is subject to the 500 step maximum.</p> <p>The step sub-element is mutually exclusive with template, instructions, variableValue, skills, and weight.</p>	Required.	Recursive step definition as defined by this table.	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps

Element name	Description	Required or optional	Type	Supported attributes
runAsUser	<p>User ID under which a particular step is to be performed. The value that you specify on the runAsUser element is considered to be the <i>runAsUser ID</i> for the step. When a runAsUser is not specified for a step, the step is performed under the step owner user ID.</p> <p>The runAsUser element is required if the approver element is specified or the stepSignature element is specified.</p>	Optional.	nonNullString	<p>The following attribute is supported for the <i>runAsUser</i> sub-element:</p> <p>substitution Indicates whether the runAsUser value contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>If you use variable substitution, understand that the variable must be an instance variable; it cannot be a global variable.</p> <p>If you want to use the element <i>stepSignature</i>, the sub-element <i>substitution</i> is always <i>false</i>.</p>
approver	<p>A user ID, or a list of user IDs separated by blanks. At least one user ID must approve the step before it is performed on behalf of the user ID that is specified with the runAsUser element.</p> <p>To specify multiple required approvers, use multiple approver elements (up to 12). The approver element is optional. If it is specified, the runAsUser element is required.</p> <p>The sub-element approver is mutually exclusive with the sub-element stepSignature.</p>	Optional	approverType	<p>The following attribute is supported for the <i>approver</i> sub-element:</p> <p>substitution Indicates whether the approver user IDs contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
stepSignature	The signature status for a runAsUser step. The sub-element approver is mutually exclusive with the sub-element stepSignature.	Optional	stepSignatureType	<p>The following attributes are supported for the <i>stepSignature</i> sub-element:</p> <p>Signer A valid z/OS user ID or valid z/OS group ID. A user ID or a list of user IDs separated by blanks. At least one user ID must sign the step before it is performed on behalf of the user ID that is specified with the runAsUser element. To specify multiple required signers, use multiple signer elements. Up to 12 signer elements are supported. The signer element is optional. If it is specified, the runAsUser element is required.</p> <p>Signature The signature that is generated after the signer signs for the step in Workflow Editor. The use of the signature element requires that both the runAsUser and signer elements are specified. It is omitted if the signer element is not specified.</p> <p>The following attribute is supported for the <i>signature</i> sub-element:</p> <p>version Indicates the version of the signature.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
condition	A conditional step.	Optional.	Condition for performing this step.	<p>The following attributes are supported for the <i>condition</i> sub-element:</p> <p>expression Provides an expression based on input variables (global or instance) and Boolean logic. When the expression resolves to <i>true</i> for the current workflow instance, the step can be performed. An expression is required, and must have the data type <i>conditionType</i>.</p> <p>description Describes the condition that must be satisfied before the step can be performed. A description is required.</p> <p>targetState The state to which the step is set when the expression evaluates to <i>true</i>. The valid values are <i>Ready</i> and <i>Skipped</i>. The target state is optional. If not specified, the default is <i>Ready</i>.</p>
suspend	<p>A suspended step. When automation processing reaches a step that includes the suspend element, automation stops at the step. Here, z/OSMF can send an email notification to one or more recipients that you specify, such as a person who should be prompted to take action.</p> <p>With the suspend element, you can define an email notification by using the following sub-elements of suspend:</p> <ul style="list-style-type: none"> • toRecipients • subject • content 	Optional.	suspendType	No attributes are supported for this sub-element.
toRecipients	For a suspended step, specifies the email addresses of the persons to be notified of the suspended step. If you omit this value, no email is sent.	Optional.	velocityString	No attributes are supported for this sub-element.
subject	For a suspended step, specifies a subject for the notification email. If you omit this value, the email subject is set to no subject by default.	Optional.	velocityString	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
content	For a suspended step, specifies the text of the message for the notification email. If you omit this value, the email content is set to no content by default.	Optional.	velocityString	No attributes are supported for this sub-element.
variableValue	References a variable that is defined earlier. The variableValue sub-element is mutually exclusive with the step sub-element.	Optional.	Empty	<p>The following attributes are supported for the <i>variableValue</i> sub-element:</p> <p>name The name of the referenced variable. The name is required, and must be a string consisting of letters (uppercase or lowercase), numeric digits, and the underscore character. This value must begin with a letter.</p> <p>The name and scope combination must refer to a defined variable.</p> <p>scope The scope of the referenced variable. The scope is optional. If specified, the value must be <i>instance</i> or <i>global</i>. The default is <i>instance</i>.</p> <p>The name and scope combination must refer to a defined variable.</p> <p>required Whether the variable must have a value for this step. The required attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>When set to <i>true</i>, the Workflows task does not allow the user to complete the step without providing a value (if a default is not defined in the XML).</p> <p>noPromptIfSet Whether the variable widget is displayed in read-only mode, if the variable already has a value. The noPromptIfSet attribute is an optional, Boolean value. If not specified, the default is <i>false</i>, that is, always display the variable widget in read/write mode.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
instructions	Detailed documentation on what the user must do to perform the step. This sub-element is mutually exclusive with the step sub-element.	Required.	nlsRichVelocityString	The following attribute is provided for the <i>instructions</i> sub-element: substitution Indicates whether instructions contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
weight	The relative difficulty of the step compared to other steps within this workflow. The weight sub-element is mutually exclusive with step sub-element.	Required.	Integer value 1 - 1000.	No attributes are supported for this sub-element.
skills	The type of skills that are required to perform this step. The skills sub-element is mutually exclusive with step sub-element.	Optional.	nlsString	No attributes are supported for this sub-element.
autoEnable	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>autoEnable</i> is not specified, the default is <i>false</i> .	Optional.	Boolean	No attributes are supported for this sub-element.
canMarkAsFailed	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. If <i>canMarkAsFailed</i> is not specified, the default is <i>false</i> . When set to <i>true</i> , the Review Instructions page in the Step Perform wizard includes the option to allow the step owner to mark the step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user.	Optional.	Boolean	No attributes are supported for this sub-element.
rest	Identifies the step as a REST step.	Optional.	String.	The sub-elements and attributes of a REST step are described in Table 594 on page 1206 .

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
template	<p>Identifies the step as a <i>template step</i>, which is a step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script.</p> <p>The template sub-element is mutually exclusive with the step sub-element.</p>	Optional.	<p>The template sub-element must contain one of the following sub-elements:</p> <p>inlineTemplate A file template or executable template that is specified in the workflow definition file. The value must be type <code>velocityString</code>.</p> <p>fileTemplate Path name of the external file that contains the template. The contents of the file are treated as type <code>velocityString</code>.</p>	<p>The following attribute is supported for both the <code>inlineTemplate</code> and <code>fileTemplate</code> sub-elements:</p> <p>substitution Indicates whether the template contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>. A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
submitAs	Indicates the type of executable program.	Optional.	<p>"TSO-REXX" Run a REXX exec program in real time.</p> <p>"TSO-UNIX-REXX" Run a REXX exec program for the UNIX environment in real time.</p> <p>"TSO-UNIX-shell" Run a UNIX shell script in real time.</p> <p>"JCL" Submit a JCL job for batch processing on z/OS. The results are indicated in the job log.</p> <p>"TSO-REXX-JCL" Submit a JCL job that contains a REXX program. The program runs as a batch job on z/OS; the results are indicated in the job log.</p> <p>"shell-JCL" Submit a JCL job that contains a UNIX shell script. The program runs as a batch job on z/OS; the results are indicated in the job log.</p> <p>A REXX exec that is written to be run in a UNIX shell environment should be submitted as "TSO-UNIX-shell" or "shell-JCL".</p>	<p>The following attribute is supported for the <i>template</i> element when the <i>submitAs</i> sub-element is "JCL", "TSO-REXX-JCL", or "shell-JCL":</p> <p>maxRc Maximum return code value to consider successful. This attribute is optional. If specified, the value must be an integer in the range 0 - 4095. If not specified, the default is 0.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
maxlrecl	<p>For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job.</p> <p>This value is used when the step is performed automatically (<i>autoEnable=true</i>). If the step is performed manually, the user can optionally specify the maximum record length on the Edit JCL page in the Workflows task.</p>	Optional.	Integer value 80 - 1024. The default is 1024 bytes.	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
output	<p>For a step that creates a properties file, this element specifies the default name and location of the file. The properties file can be a data set, a UNIX file, or a JES spool file.</p> <p>The Workflows task allows the user to modify the file name and location, as needed.</p> <p>For more information about the properties file, see “Creating a properties file” on page 1124.</p>	Optional.	String.	<p>The following attributes are supported for the <i>output</i> sub-element:</p> <p>substitution Indicates whether the properties file name uses variable substitution. This is an optional, Boolean value. The default is <i>false</i>.</p> <p>needResolveConflicts Indicates whether the user is prompted to resolve variable conflicts from the properties file. This is an optional, Boolean value. The default is <i>true</i>.</p> <p>If variable conflicts exist, the user is prompted to resolve the conflicts. A value of <i>true</i> must be specified for the Workflows task to display the variables in the Input Variables page. If set to <i>false</i>, the Workflows task uses the properties file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden by variables in the properties file.</p> <p>sysoutDD= Indicates whether the properties file is a JES spool file that is produced by a job step in the executed job. This is an optional, Boolean value. The default is <i>false</i>, which means that the properties file is either a data set or a UNIX file, as specified on the output element. If set to <i>true</i>, the properties file is a JES spool file. The location that is specified on the output element must identify a valid DD name and, optionally, a job step name.</p> <p>loadOutputFileArrayValue= Indicates the behavior for handling variable conflicts. If set to <i>true</i>, the workflow uses the array variable values from the output file, rather than from the Workflows task. If set to <i>false</i>, the workflow uses the existing values from the Workflows task.</p>
successPattern	Regular expression that is returned for a successful program execution. This element is required. You must specify one (and only one) regular expression for a successful program completion.	Required.	String.	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
failedPattern	Optional regular expression that can be returned for program execution failures. You can omit this element or specify up to 100 different specifications for <i>failedPattern</i> . This property might be null.	Optional.	String.	No attributes are supported for this sub-element.
outputVariablesPrefix	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.	Optional.	String.	<p>The following attribute is supported for the <i>outputVariablesPrefix</i> sub-element:</p> <p>needResolveConflicts Indicates whether the user is prompted to resolve variable conflicts from the program output variables. This attribute is an optional, Boolean value. If not specified, the default is <i>true</i>.</p> <p>If variable conflicts exist, the user is prompted to resolve the conflicts. A value of <i>true</i> must be specified for the Workflows task to display the variables in the Input Variables page.</p> <p>If set to <i>false</i>, the Workflows task uses the output file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden by variables in the output file.</p> <p>loadOutputFileArrayValue= Indicates the behavior for handling variable conflicts. If set to <i>true</i>, the workflow uses the array variable values from the output file, rather than from the Workflows task. If set to <i>false</i>, the workflow uses the existing values from the Workflows task.</p>
scriptParameters	<p>For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.</p> <p>The following sub-elements are included on the <i>scriptParameters</i> element:</p> <ul style="list-style-type: none"> <i>description</i> <i>value</i> 	Optional.	String.	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
description	Text description of the parameter in the <i>scriptParameters</i> element, such as its intended use or recommended value.	Required if the <i>scriptParameters</i> element is specified.	String.	No attributes are supported for this sub-element.
value	Value of the parameter in the <i>scriptParameters</i> element.	Required if the <i>scriptParameters</i> element is specified.	String.	No attributes are supported for this sub-element.
procName	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.	Optional.	String.	The following attribute is supported for the <i>procName</i> sub-element: substitution Indicates whether the procedure name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
regionSize	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value is specified for the step, the default is 50000.	Optional.	Integer value 50000 - 2096128.	No attributes are supported for this sub-element.
timeout	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.	Optional.	Integer value 60 - 3600.	No attributes are supported for this sub-element.
saveAsDataset	Data set name (fully qualified, no quotations) specifying where to save the file after the user edits it. When a file is generated, the presence of this element results in the <i>save as data set</i> option being presented to the user, primed with the element value, if specified. When a program is run, this element can be used as the default value. However, the Workflows task widget is always displayed to the user.	Optional.	velocityFileString	The following attribute is supported for the <i>saveAsDataset</i> sub-element: substitution Indicates whether the data set name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
saveAsUnixFile	Path name that specifies where to save the file after the user edits it. When a file is generated, the presence of this element results in the <i>save as UNIX file</i> option being presented to the user, primed with the element value, if specified. When a program is run, this element can be used as the default value. However the Workflows task widget is always displayed to the user.	Optional.	velocityFileString	The following attribute is supported for the <i>saveAsUnixFile</i> sub-element: substitution Indicates whether the path name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
predefinedVariable	For a step that submits a job, this sub-element sets a variable in the body of the job. A predefined variable is treated as a string substitution. The substitution applies to the current step only. You can specify multiple predefined variables per step.	Optional.	predefinedVariableType	The following attribute is supported for the <i>predefinedVariable</i> sub-element: name Name of the predefined variable. The name is required, and must be of data type string. To avoid overriding the variables that are defined for the workflow, use a unique name for the predefined variable.
feedbackItem	Optionally includes a feedback form for the step in the Workflows task with questions for the step owner to answer. Up to 100 feedback items (questions) can be specified for a step.	Optional.	feedbackItemType	The following attributes are supported for the <i>feedbackItem</i> sub-element: name Name of the feedback question. The name is required, and must be of data type string. required Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
question	The question to be asked (a text string). For example: <i>How difficult was this step?</i>	Required, if the element <i>feedbackItem</i> is specified.	String.	No attributes are supported for this sub-element.

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
answers	<p>Answer format, which is defined by including one of the following sub-elements:</p> <p>singleSelect One answer (and only one) can be selected from the available choices.</p> <p>multipleSelect More than one answer can be selected from the available choices.</p> <p>text No choices are provided; the user answers the question by entering a text value.</p>	Required, if the element <i>feedbackItem</i> is specified.	answersType	No attributes are supported for this sub-element.
singleSelect	A sub-element of the element <i>answers</i> . Indicates that one answer (and only one) can be selected from the available choices that are listed in <i>answers</i> .	Optional.	selectType	<p>The following attribute is supported for the <i>multipleSelect</i> sub-element:</p> <p>hasOtherAnswer Displays the choice "Other" as a selectable answer. Is displayed last in a list of multiple choices.</p>
label	One or more answers to be listed as selectable choices for the question. At least one label (answer) must be specified, up to a maximum of 50.	Required.	labelType	<p>The following attributes are supported for the <i>label</i> sub-element:</p> <p>value Answer that can be selected. Required.</p> <p>required Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p>
multipleSelect	A sub-element of the element <i>answers</i> . Indicates that more than one answer can be selected from the available choices.	Optional.	selectType	<p>The following attribute is supported for the <i>multipleSelect</i> sub-element:</p> <p>hasOtherAnswer Displays the choice "Other" as a selectable answer. Is displayed last in a list of multiple choices.</p>

Table 593. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
label	One or more answers to be listed as selectable choices for the question. At least one label (answer) must be specified, up to a maximum of 50.	Required.	labelType	The following attributes are supported for the <i>label</i> sub-element: value Answer that can be selected. Required. required Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
text	A sub-element of the element <i>answers</i> . Indicates that no choices are provided; the user answers the question by entering a text value.	Optional.	String.	No attributes are supported for this sub-element.

Table 594. Step elements summary table: Additional sub-elements for REST steps

Element name	Description	Required or optional	Type	Supported attributes
httpMethod	Indicates the HTTP method that is used for issuing the REST request. The following values are valid: <ul style="list-style-type: none"> • GET • PUT • POST • DELETE. 	Required.	String	None.
schemeName	Scheme name that is associated with the REST request. If specified, this element must be set to "http."	Optional.	String	None.
hostname	Host name or IP address of the system to which the REST request is directed. For example: <code>www.ibm.com</code> .	Optional	String	The following attribute is supported for the hostname sub-element: substitution Indicates whether the host name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
port	Port number to use for the REST request.	Optional.	String	The following attribute is supported for the port sub-element: substitution Indicates whether the port contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.

Table 594. Step elements summary table: Additional sub-elements for REST steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
uriPath	URI path to use for the REST request.	Required.	String	<p>The following attribute is supported for the uriPath sub-element:</p> <p>substitution Indicates whether the URI path contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
queryParameters	For a GET or POST request, this element contains the query parameters.	Optional.	String	<p>The following attribute is supported for the queryParameters sub-element:</p> <p>substitution Indicates whether the query parameters contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
requestBody	For a PUT or POST request, this element contains the request body.	Optional.	String	<p>The following attribute is supported for the requestBody sub-element:</p> <p>substitution Indicates whether the request body contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
expectedStatusCode	The expected HTTP status code from the REST API request. If this value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a job template step returns a return code that is greater than the allowed maximum return code.	Required.	Integer	None

Table 594. Step elements summary table: Additional sub-elements for REST steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
actualStatusCode	The actual HTTP status code that is received from the REST request. To obtain this value, map it to a workflow variable.	Optional	Integer	None
propertyMapping	The property from the REST response body that is mapped to a workflow variable. You can specify multiple <code>propertyMapping</code> elements in a REST step.	Optional	String	<p>The following attributes are supported for the <code>propertyMapping</code> sub-element:</p> <p>mapFrom Specifies the supplied value to be mapped. This attribute is optional. If specified, the type must be <code>nonNullString</code>.</p> <p>mapTo Specifies the variable to which the supplied value is to be mapped. This attribute is optional. If specified, the type must be <code>nonNullString</code>.</p>

Table 595. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow)

Element name	Description	Required or optional	Type	Supported attributes
variableMapping	Used to transfer variable values between the called workflow and calling workflow.	Optional.	Any	No attributes are supported for this sub-element.
fromCallingToCalled	Used to transfer variable values from the calling workflow to the called workflow.	Optional.		No attributes are supported for this sub-element.
regExpression	Used for filtering on variable names. Specify a portion of the variable name with one or more wildcard characters.	Optional.	nonNullString	No attributes are supported for this sub-element.
variableName	Name of the variable.	Optional.	nonNullString	The following attribute is supported for the <i>variableName</i> sub-element: mapTo Specifies the variable to which the supplied value is to be mapped. This attribute is optional. If specified, the type must be nonNullString.
fromCalledToCalling	Used to transfer variable values from the called workflow to the calling workflow.	Optional		The following attribute is supported for the <i>fromCalledToCalling</i> sub-element: override Indicates whether the variable settings in this element are to take precedence over the variables in the calling workflow. The default is false. This attribute is optional. If specified, it must be a Boolean value.
regExpression	Used for filtering on variable names. Specify a portion of the variable name with one or more wildcard characters.	Optional.	nonNullString	No attributes are supported for this sub-element.
variableName	Name of the variable.	Optional.		No attributes are supported for this sub-element.
callingStepWeight	The relative difficulty of the step compared to other steps within this workflow. The <i>callingStepWeight</i> sub-element is mutually exclusive with the <i>step</i> sub-element.	Required.	Integer value 1 - 1000.	No attributes are supported for this sub-element.

Table 595. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow) (continued)

Element name	Description	Required or optional	Type	Supported attributes
callingStepSkills	The type of skills that are required to perform this step. The <i>callingStepSkills</i> sub-element is mutually exclusive with the <i>step</i> sub-element.	Optional.	nlsString	No attributes are supported for this sub-element.
callingStepAutoEnable	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>callingStepAutoEnable</i> is not specified, the default is <i>false</i> .	Optional.	Boolean	No attributes are supported for this sub-element.
canCallingStepMarkAsFailed	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. If <i>canCallingStepMarkAsFailed</i> is not specified, the default is <i>false</i> . When set to <i>true</i> , the Review Instructions page in the Step Perform wizard includes the option to allow the step owner to mark a step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user.	Optional.	Boolean	No attributes are supported for this sub-element.
calledWorkflowDefinitionFile	Path name of an external file that contains the workflow definition for the called workflow. Can be a fully qualified path name, or a relative path name (that is, relative to the location of the calling workflow). For a relative path, the path must begin with <i>./</i> or <i>../</i> . After beginning with this notation, all other instances of <i>./</i> or <i>../</i> in the path will be resolved.	Optional.	Path name to template file, the contents of which are treated as type <i>velocityString</i> .	No attributes are supported for this sub-element.
calledWorkflowDescription	A more detailed description of the step.	Required.	nlsRichString	No attributes are supported for this sub-element.
calledWorkflowID	The name of the workflow. The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task. You can use the calledWorkflowID and calledWorkflowVersion to identify a called workflow.	Optional.	String consisting of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter.	No attributes are supported for this sub-element.

Table 595. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow) (continued)

Element name	Description	Required or optional	Type	Supported attributes
calledWorkflowVersion	<p>The version of this workflow definition file. Update this value whenever you change any portion of the workflow definition file, including changes to the primary XML file or any sub-files or referenced files.</p> <p>The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.</p> <p>The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task. You can use the calledWorkflowID and calledWorkflowVersion to identify a called workflow.</p>	Optional.	nonNullString	No attributes are supported for this sub-element.
calledWorkflowMD5	An MD5 encrypted value (a 128-bit hash value) that you can use to identify the called workflow.	Optional.	nonNullString	No attributes are supported for this sub-element.

Variable definition elements and types summary

Table 596. Variable definition elements summary				
Element name	Description	Required or optional	Type	Supported Attributes
variable	Contains the definition of a variable. Up to 1500 variables can be defined in a workflow.	Optional.	<p>If specified, the <i>variable</i> element can contain the following sub-elements:</p> <ul style="list-style-type: none"> • label (required) • abstract (required) • description (required) • exposeToUser (optional) • category (required) • datastore (optional) <p>And, the <i>variable</i> element must contain at least one of the type-specific sub-elements, which are listed in Table 597 on page 1217.</p>	<p>The following attributes are supported for the <i>variable</i> element:</p> <p>name The name of the variable. The name is required, and must be a string consisting of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter.</p> <p>The combination of name and scope must be unique within the workflow.</p> <p>scope The scope of the variable. The scope is required, and the value must be <i>instance</i> or <i>global</i>. The default is <i>instance</i>.</p> <p>The combination of name and scope must be unique within the workflow.</p> <p>visibility Specifies whether the variable is intended for public or private use. This attribute is intended for the workflow author's use. The visibility setting does not affect how the variable is processed by the Workflows task. This attribute is optional. The default is <i>private</i>.</p>
label	A short label for the UI widget.	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
abstract	A brief description of the variable for the UI widget.	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
description	A longer explanation of what the variable is used for, and perhaps what the syntactic requirements are.	Required if the variable element is specified.	nlsRichString	No attributes are supported for this sub-element.

Table 596. Variable definition elements summary (continued)

Element name	Description	Required or optional	Type	Supported Attributes
exposeToUser	For a step that runs a JCL job. If included, the exposeToUser element indicates that the variable is to be included in the List variables for substitution window of the Workflows task. This element supports both global variables and instance variables in Workflows task. If this element is not specified, the variable is not shown in the Edit JOB statement page, and thus, cannot be selected by the user for the substitution of another value in the JOB statement.	Optional. Applicable only when the variable element is specified.	exposeToUserType	usage Specify the purpose of the variable. This text is displayed next to the variable in the Edit JOB Statement window of the Workflows task. The usage is optional. If specified, its data type must be an nlsString.
category	The name of the logical grouping to which this variable belongs. The default is <i>general</i> .	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
datastore	Place where the variable value is stored over time. Because z/OSMF is the only supported location for variable values, it is not necessary to specify the datastore element.	Optional. Applicable only when the variable element is specified.	A single, empty, required zOSMF sub-element	No attributes are supported for this sub-element.

Table 597. Variable definition type-specific sub-elements summary			
Element name	Description	Type	Supported attributes
Boolean	Boolean type	Empty	No attributes are supported for this sub-element.
default	Default value	Boolean. The default is <i>true</i> .	No attributes are supported for this sub-element.
string	String type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minLength Minimum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the maxLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the validationType and regularExpression sub-elements.</p> <p>maxLength Maximum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the minLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the validationType and regularExpression sub-elements.</p> <p>validationType Validation types. This sub-element is optional. If specified, it must have one of the following values: ALPHA, ALPHAB, ALPHANUM, BIT, DSMEMBERNAME, DSNNAME, GROUP, HEX, IPADDR, IPADDR4, IPADDR6, TSUSERID, UNIXID, USERID, VOLSER.</p> <p>The validationType sub-element mutually excludes the minLength and maxLength combination and the regularExpression sub-element.</p> <p>The UNIXID validation type verifies that a z/OS UNIX UID or GID is in the range 0 – 2147483647. Here, a UID or GID is treated as a string, not an integer. If you have code that treats a UID or GID as numeric, use an integer type to define the variable, instead of a string validation type. You can enforce the minimum and maximum values within the integer variable definition.</p> <p>regularExpression Standard regular expression that constrains the variable value. This sub-element is optional. If specified, it must have the <code>bigNonNullString</code> data type.</p> <p>The <code>regularExpression</code> sub-element mutually excludes the <code>minLength</code> and <code>maxLength</code> combination and the <code>validationType</code> sub-element.</p> <p>errorMessage Overrides the default error message for an incorrect value. This sub-element is optional. If specified, it must have an <code>nlString</code> data type.</p> <p>choice Added to the choice list. This sub-element is required if <i>valueMustBeChoice=true</i>.</p> <p>Up to 1337 choices are allowed, and each choice must adhere to any <code>minLength</code>, <code>maxLength</code>, <code>validationType</code>, and <code>regularExpression</code> specified.</p> <p>default Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be an <code>nlUnboundedString</code> that adheres to any <code>minLength</code>, <code>maxLength</code>, <code>validationType</code>, and <code>regularExpression</code> specified.</p> <p>If <i>valueMustBeChoice=true</i>, the default value must be one of the choice values specified.</p>	<p>The following attributes are supported for the <i>string</i> sub-element:</p> <p>multiline Specifies a single-line text box widget or a multi-line text box. This attribute is optional, and its data type is Boolean. If unspecified, its value is <i>false</i> (default).</p> <p>valueMustBeChoice Indicates whether the value must come from the provided choices. This attribute is optional, and its data type is Boolean. If unspecified, its value is <i>false</i> (default).</p> <p>If set to <i>true</i>, at least one <i>choice</i> sub-element must be specified.</p>
integer	Integer type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minValue Minimum value. This sub-element is optional. If specified, its value must be an integer.</p> <p>If the <code>maxValue</code> is specified, the <code>maxValue</code> must be greater than or equal to the <code>minValue</code>.</p> <p>maxValue Maximum value. This sub-element is optional. If specified, its value must be an integer.</p> <p>If the <code>minValue</code> is specified, the <code>maxValue</code> must be greater than or equal to the <code>minValue</code>.</p> <p>default Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be an integer that adheres to any <code>minValue</code> and <code>maxValue</code> specified.</p>	No attributes are supported for this sub-element.

Table 597. Variable definition type-specific sub-elements summary (continued)			
Element name	Description	Type	Supported attributes
decimal	Decimal type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minValue Minimum value. This sub-element is optional. If specified, its value must be a decimal. If the maxValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>maxValue Maximum value. This sub-element is optional. If specified, its value must be a decimal. If the minValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>default Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be a decimal that adheres to any minValue and maxValue specified. Decimal places are rounded by the Workflows task, based on the decimalPlaces attribute value.</p>	<p>The following attribute is supported for the <i>decimal</i> sub-element:</p> <p>decimalPlaces Maximum number of decimal places that can be specified. The value can be an integer in the range of 1 - 6. The default is 1.</p>
time	Time type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minValue Minimum value. This sub-element is optional. If specified, its value must be the time in <i>hh:mm:ss</i> format. If the maxValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>maxValue Maximum value. This sub-element is optional. If specified, its value must be the time in <i>hh:mm:ss</i> format. If the minValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>default Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be the time in <i>hh:mm:ss</i> format and must adhere to any minValue and maxValue specified.</p>	No attributes are supported for this sub-element.
date	Date type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minValue Minimum value. This sub-element is optional. If specified, its value must be the date in <i>yyyy-mm-dd</i> format. If the maxValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>maxValue Maximum value. This sub-element is optional. If specified, its value must be the date in <i>yyyy-mm-dd</i> format. If the minValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p>default Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be the date in <i>yyyy-mm-dd</i> format and must adhere to any minValue and maxValue specified.</p>	No attributes are supported for this sub-element.

Table 597. Variable definition type-specific sub-elements summary (continued)			
Element name	Description	Type	Supported attributes
password	Password type.	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p>minLength Minimum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the maxLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the regularExpression sub-element.</p> <p>maxLength Maximum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the minLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the regularExpression sub-element.</p> <p>regularExpression Standard regular expression that constrains the variable value. This sub-element is optional. If specified, it must have the bigNonNullString data type.</p> <p>The regularExpression sub-element mutually excludes the minLength and maxLength sub-elements.</p> <p>errorMessage Overrides the default error message for an incorrect value. This sub-element is optional. If specified, it must have an nlsString data type.</p>	No attributes are supported for this sub-element.
array	Array type	None; no sub-elements are applicable.	No attributes are supported for this sub-element.

atCreate element summary

Table 598. *atCreate* element summary

Element name	Description	Required or optional	Type	Supported Attributes
atCreate	For users of the Create Workflow REST service, the <i>atCreate</i> element provides more options for working with variables. Up to 1500 <i>atCreate</i> elements can be defined in a workflow.	Optional.	<i>atCreateType</i>	<p>The following attributes are supported for the <i>atCreate</i> element:</p> <p>name Specifies the variable for which the variable attributes are being set. This attribute is required.</p> <p>scope Specifies the scope of the variable. This value is set to <i>instance</i> (the only valid value) or is omitted. The default is <i>instance</i>.</p> <p>required For a workflow that is created through the Create Workflow REST service, this attribute indicates whether the variable must be set to a value at the time of workflow creation. This attribute is optional; the default is <i>false</i>. If a variable is marked as "required," but the variable is not given a value, an attempt to create the workflow through the Create Workflow REST service fails with an error.</p> <p>For a workflow that is created through the Workflows task user interface, this option is ignored.</p> <p>prompt For users of the Create Workflow REST service, this attribute identifies a variable that <i>should</i> be prompted for by the program that issues the REST service. By itself, the prompt attribute does not enforce any behavior for the workflow creation. However, by setting prompt to <i>true</i>, you can indicate that prompting is recommended for the variable. The user of the Create Workflow REST service can query the value of the prompt attribute for any variables in the workflow to determine whether any variables should be prompted for. This attribute is optional; the default is <i>false</i>.</p>

Chapter 3. Creating your own z/OSMF plug-ins

z/OSMF provides a modular framework that you can use to bring together all of your z/OS system management applications. z/OSMF supports different levels of integration that range from adding a resource link to adding an application link to create your own z/OSMF plug-ins.

To help you decide which integration method is best for you, consider the following recommendations:

- If you already have a web application that does not and is not expected to have natural integration points with existing z/OSMF tasks or applications, use the Links task to add your web application as a link in z/OSMF.
 - With the link integration, you do not need to create a new application. Minimal or no code change is needed, and you can consolidate your application to the z/OSMF ecosystem.
- If you are about to create a new web application and want to use the z/OSMF security controls, persistence capabilities, and REST services, it is recommended to create your own web-based application and import it as a third-party plug-in on z/OSMF.
 - If your web application can use the z/OSMF REST services to complete business logic, only the web-based application a needed.
 - If your web application requires a function that is not provided in z/OSMF or you would like to reuse your existing TSO program, it is recommended to create a backend module. You can also enhance your existing module to interact with the web application through z/OSMF TSO/E REST services.
- The benefits of using a plug-in integration are as follows:
 - You do not need to have an application server and associated configuration.
 - You can leverage the existing z/OSMF security controls and high availability architecture.
 - You can use the REST services that include functions like data persistence, multi-system communication, handling of data sets, Unix files, and Jobs.
 - You can use the z/OSMF JavaScript interfaces that improve user experience. This includes data set and file selector, the modern VS-code based editor that can view data sets and files, file comparison, and so on.
 - You can use z/OSMF application linking to interact with other z/OSMF tasks or websites.
 - With your own backend TSO module, you can use existing native modules to provide data for the web application without reconstructing it with Java or C methods.
- No matter which approach you take to integrate with z/OSMF, you can use z/OSMF Application Linking Manager to launch other z/OSMF tasks from your application for richer functions and a better user experience. This includes when your web application is a stand-alone and does not run in a z/OSMF window,

Table 599. Summary of benefits for each integration type			
Integration type	Best fit	Benefits	How
Link	You already have a web application that does not and is not expected to have natural integration points with existing z/OSMF tasks or applications.	You don't need to create a new application. Minimum code change is needed. You can consolidate your application to the z/OSMF ecosystem.	Use the Links task to add the resource as a link in z/OSMF.

Table 599. Summary of benefits for each integration type (continued)

Integration type	Best fit	Benefits	How
A third-party plug-in	You are about to create a new web application and want to leverage the z/OSMF security controls, high availability, persistence capabilities, and the variety of REST services.	<p>You do not need to have an application server and associated configuration.</p> <p>You can leverage z/OSMF security controls and high availability architecture.</p> <p>You can use REST services such as data persistence, multi-system communication, handling of data sets, Unix files, and Jobs.</p> <p>You can use the z/OSMF JavaScript interfaces that improve user experience. This includes data set and file selector, modern VS-code based editor that is used to view data sets and files, file comparison, and so on.</p> <p>You can use z/OSMF application linking to interact with z/OSMF tasks or other websites.</p> <p>With your own backend TSO module, you can have existing native modules provide data for the web application without reconstructing it to Java or C methods.</p>	<p>Create your own web application.</p> <p>Create a property file that represents the plug-in.</p> <p>(Optional) Create a backend module.</p> <p>Use z/OSMF Import Manager task to add the application to z/OSMF.</p>
Application Linking	You have a web application that can launch or be launched by other z/OSMF tasks or web-based applications and display specific context.	Interact with other z/OSMF tasks to provide rich functions and user experience.	<p>Send the event to use existing z/OSMF event handlers, such as "view job status".</p> <p>Or use the z/OSMF application linking task to define an event handler and provide a context-based function for other tasks to launch.</p>

Process of joining the z/OSMF ecosystem

The z/OSMF framework provides the infrastructure, security, and services you need to create the web-based applications to be included in your plug-ins.

Third-party application

In z/OSMF, a plug-in is a collection of one or more web-based applications (referred to as tasks) that add function to z/OSMF. z/OSMF ships with several plug-ins. For more information, see [Selecting which z/OSMF services to add in IBM z/OS Management Facility Configuration Guide](#).

In addition to the shipped plug-ins, z/OSMF allows you to create your own plug-ins to add installation-specific functions to z/OSMF. The process of adding your own plug-ins to z/OSMF includes the following activities:

1. Developing a web-based application and the supporting documentation for the functions you want to add to z/OSMF.

For z/OSMF programming interfaces that can enrich your functions, see the following links.

- a. [“Using the z/OSMF core JavaScript APIs” on page 1227](#)
 - b. [“Using the Application Linking Manager JavaScript APIs” on page 1240](#)
 - c. [“Logging client messages in the z/OSMF log” on page 1251](#)
 - d. [“Retrieving files and resources for your application” on page 1256](#)
 - e. Chapter 1, “Using the z/OSMF REST services,” on page 1
 - f. [“Example of a third party z/OSMF plug-in” on page 1226](#)
2. Storing the application and its documentation in the UNIX file system, and setting 644 permissions for files and 755 permissions for folders.
 3. Creating a property file in the UNIX file system that defines the parameters that are required for z/OSMF to configure your plug-in.
 4. Using the z/OSMF Import Manager task to import the property file.
 5. Setting up security for your plug-in. After which, you must refresh the security management product on your system and restart the z/OSMF server to have your changes take effect.

Link

In z/OSMF, a link is a bookmark that adds a URL to z/OSMF and displays an icon on the z/OSMF user interface. A z/OSMF administrator decides whether users can open it in a separate browser tab or in a z/OSMF window. If the link is defined to open in a z/OSMF window, the web application with the URL is rendered in a z/OSMF window when the user clicks the icon, just like with other z/OSMF plug-ins. The process of adding your own web application as a link to z/OSMF includes the following activities:

1. Developing a web-based application or having an existing web-based application.

If the application opens as a z/OSMF window, the following services can be used to provide a better user experience.

- a. [“Using the z/OSMF core JavaScript APIs” on page 1227](#)
 - b. [“Using the Application Linking Manager JavaScript APIs” on page 1240](#)
 - c. Chapter 1, “Using the z/OSMF REST services,” on page 1
2. Using the z/OSMF Links task to define the URL of the application, and whether to open in a browser tab or z/OSMF window.
 3. Setting up security for the link.

Application linking

Tasks in z/OSMF can interact with one another with the application linking for a smoother user experience and richer functions.

If your web application expects to launch other z/OSMF tasks, the process includes the following activities:

- In your web application, send a registered event with the application linking JavaScript APIs. For more information, see [“Using the Application Linking Manager JavaScript APIs” on page 1240](#).

If your web application can provide common functions that can be launched by other z/OSMF tasks, the process includes the following activities:

1. Define an event type and an event handler in Application Linking Manager task, or use a properties file to import with Import Manager task.
2. In your web application, listen to the event and process the event when you receive it. For more information, see [“Using the Application Linking Manager JavaScript APIs” on page 1240](#).

Example of a third party z/OSMF plug-in

The z/OSMF framework provides the infrastructure, security, and services you need to create the web-based applications to be included in your plug-ins.

For an example of how to create and deploy your own z/OSMF third party plug-in, see the example in GitHub. It contains a user interface that is based on the popular Angular framework, and uses several z/OSMF Representational State Transfer (REST) APIs to perform operations on a z/OS host system.

The sample plug-in is provided here: [z/OSMF third party plug-in example in GitHub \(github.com/IBM/IBM-Z-zOS/tree/master/%20zOSMF/ExternalPluginExample-TSOBackend\)](#).

Developing web-based applications

The z/OSMF framework provides the infrastructure, security, and services you need to create the web-based applications to be included in your plug-ins.

Specifically, z/OSMF provides the following resources:

JavaScript APIs

z/OSMF core provides application programming interfaces (APIs) that allow you provide a smoother and modern user experience, such as Unix file and data set selector, modern file editor, comparison between two files, . For more details, see [“Using the z/OSMF core JavaScript APIs” on page 1227](#) and [“Using the Application Linking Manager JavaScript APIs” on page 1240](#).

REST Services

z/OSMF provides several REST interfaces, which you can use to simplify the creation of your z/OSMF plug-ins. For more details, see [Chapter 1, “Using the z/OSMF REST services,” on page 1](#).

Client Side Logger

z/OSMF provides a client-side logging framework that you can use to route client messages to the z/OSMF log, which is the same log that is used by plug-ins that ship with z/OSMF. For more details, see [“Logging client messages in the z/OSMF log” on page 1251](#).

File Retrieval Service

z/OSMF provides a file retrieval service that you can use to specify the file for z/OSMF core to display when it launches your application and to specify any additional files or resources your application may need. For more details, see [“Retrieving files and resources for your application” on page 1256](#).

Secure Environment

z/OSMF uses the System Authorization Facility (SAF) interface on the z/OS host system to authenticate users and to grant users access to z/OS system management tasks. z/OSMF security also depends on plug-in developers identifying and remediating security vulnerabilities in their applications. For more details, see [“Securing your applications” on page 1274](#).

Requirements

Your application must:

- Be written in a markup or programming language that a web browser can interpret, such as HTML, JavaScript, and CSS. z/OSMF does not support markup or programming languages that must be interpreted by the z/OSMF server.
- Be stored in the UNIX file system with 644 permissions for files and 755 permissions for folders.
- Add new function to z/OSMF or update a plug-in that you created and previously imported. You cannot use plug-ins to modify the z/OSMF framework, the z/OSMF user interface, or other z/OSMF plug-ins.

Using the z/OSMF core JavaScript APIs

z/OSMF provides a set of JavaScript APIs to facilitate the user experience when you deal with files and data sets, and utility functions for application developers to control the application window and its state.

Functions with data sets and Unix files

Table 600. Functions with data sets and Unix files		
Function	Usage	Where it is described.
editorWithContent	Load provided content in the z/OSMF VS Code based editor.	“editorWithContent function” on page 1227
editorWithPath	Load the content of a sequential data set, a data set member, or a Unix file in the z/OSMF VS Code based editor.	“editorWithPath function” on page 1228
utilPathSelector	Launch a data set and file search window for user to select a file or a data set.	“utilPathSelector function ” on page 1229
editorCompare	Launch a comparison view in z/OSMF desktop editor of two sequential data sets, data set members or Unix files.	“editorCompare function” on page 1230
editorCompareWithContent	Launch a comparison view in z/OSMF desktop editor of provided content.	“editorCompareWithContent function” on page 1230
download	Download a sequential data set, a data set member, or a Unix file with provided code pages.	“Download function” on page 1231
upload	Upload a local file to z/OS as a sequential data set, a data set member, or a Unix file with provided code pages.	“Upload function” on page 1232

editorWithContent function

If your application has text format data for the user to review or modify in a modern VS Code based editor, consider calling the **editorWithContent** function. This loads your supplied content in the z/OSMF desktop editor and can return to your application when the user action is complete.

Invoking the function

```
window.parent.zosmf.api.editorWithContent(contentParam);
```

Where **contentParam** is an object with keys of:

Table 601. Parameters of editorWithContentfunction		
Parameter Name	Description	Required Value
id	Unique identifier of the editor window	Y
content	The text format data to display in the editor.	Y
filePath	The file path to display as a subtitle in the editor window.	Y
readOnly	If the content is read-only or editable, default value is false.	N
okCallback	The call back function to trigger after user clicks OK .	N
okLabel	The label of the OK button	N
cancelCallback	The call back function to trigger after user clicks cancel.	N
cancelLabel	The label of the Cancel button	N

The function launches the z/OSMF desktop editor window with the supplied content. The window in such case cannot be pinned to the taskbar.

Example

```
var contentParam = {id:'example editor window', content:'example content', filePath: '/usr/lpp/zosmf/workflow/izu.config.setup.xml', okCallback: (content) => {console.log(content)}};

window.parent.zosmf.api.editorWithContent(contentParam);
```

Return values

The **editorWithContent** function returns a string value of the final content to the call back function.

editorWithPath function

If your application allows user to view or modify a data set or a Unix file in a modern VS Code based editor, consider calling the **editorWithPath** function. This loads your supplied data set or Unix file in the z/OSMF desktop.

Invoking the function

```
window.parent.zosmf.api.editorWithPath(contentParam);
```

Where **contentParam** is an object with keys of.

Table 602. Parameters of editorWithPath function		
Parameter Name	Description	Required Value
name	Name of a sequential data set, a data set member, or a Unix file.	Y
volser	Identifies the volume serial to be searched for the data set.	N

The function launches the z/OSMF desktop editor window with the supplied data set name or file path.

Return values

N/A. The **editorWithPath** is a void function.

Example

```
var contentParam = {id:'ZOSMF.CLAIRE.PRMLIB(IZUPRM)'};
window.parent.zosmf.api.editorWithPath(contentParam);
```

utilPathSelector function

If your application allows users to select a data set or a Unix file with searching, consider calling the **utilPathSelector** function. This allows you to search and find without having to memorize the whole data set name or file path. You can also do a fuzzy search in the file and data set selector window.

Invoking the function

```
window.parent.zosmf.api.utilPathSelector(contentParam);
```

Where **contentParam** is an object with keys of:

Table 603. Parameters of utilPathSelector function		
Parameter Name	Description	Required Value
id	Identifier of the file and data set selector window	Y
initialValue	The start point to search from, a data set name or a file path.	N
singleSelection	Whether to allow a single selection or multi-selection, default is true for only allow single selection.	N
okCallback	The call back function to trigger after user makes a selection and clicks OK .	Y
helpLink	The link to the help documentation	N
name	The title of the data set and file selector window	N

The function launches the z/OSMF data set and file search window with the supplied initial search point, if present. It returns the user selection to the caller application.

Returned values

The **utilPathSelector** function returns a JSONObject-like value **{' selectedParm ': json-object}** to the call back function with the following keys:

results:Array[son-object]

The selection results in an array of each selection.

searchType:

The type of the selection, value is between 'file' and 'dataset'.

Example

```
var contentParam = {id:'workflow_selector', 'initialValue':'/usr/lpp/zosmf/workflow',
okCallback: (selection)=>{console.log(selection)} };
window.parent.zosmf.api.utilPathSelector(contentParam);
```

Example return

```
{
  "selectedParm": {
    "searchType": "file",
    "results": [
      {
        "name": "/usr/lpp/zosmf/workflow/izu.config.setup.xml",
        "mode": "-IWXr-xr-x",
        "size": 6832,
        "uid": 0,
        "user": "OMVSKERN",
        "gid": 1,
        "group": "OMVSGRP",
        "mtime": "2021-03-30T19:15:36",
        "owner": "OMVSKERN/OMVSGRP"
      }
    ]
  }
}
```

editorCompare function

If your application has two sequential data sets, data set members, or text format Unix files for the user to compare or reconcile in a modern VS Code based comparison editor, consider calling the **editorCompare** function. This loads the two of your supplied data set or files in each side of the comparison view in the z/OSMF desktop editor. Users can navigate through the differences and merge changes.

Invoking the function

```
window.parent.zosmf.api.editorCompare(leftParm, rightParm);
```

Where **leftParm** and **rightParm** are objects with keys of:

Table 604. Parameters of editorCompare function		
Parameter Name	Description	Required Value
name	Identifier of the file and data set's full path or name.	Y

Return values

N/A. The **editorCompare** is a void function.

Example

```
var left = {name:'CIMSSRE.R25ONLY.PARMLIB(IZUPRMWQ)'};
var right = {name:'CIMSSRE.R25ONLY.PARMLIB(IZUPRMWY)'};
window.parent.zosmf.api.editorCompare(left, right);
```

editorCompareWithContent function

If your application has text format data for the user to compare or reconcile in a modern VS Code based comparison editor, consider calling the **editorCompareWithContent** function. This loads the

two pieces of your supplied data content in each side of the comparison view in the z/OSMF desktop editor.

Invoking the function

```
window.parent.zosmf.api.editorCompareWithContent(leftParm, rightParm);
```

Where **leftParm** and **rightParm** are objects with keys of:

Table 605. Parameters of editorCompareWithContent function		
Parameter name	Description	Required value
name	Name of one side of the content.	Y
content	String format content data.	Y
readOnly	Whether the side of content is read only.	N

Return values

N/A. The **editorCompareWithContent** is a void function.

Example

```
var left = {name:'Policy A', content:'left', readOnly:true };
var right = {name:'Policy B', content:'right', readOnly:true};
window.parent.zosmf.api.editorCompareWithContent(left, right);
```

Download function

If your application allows user to download a sequential data set, a data set member, or a Unix file, consider calling the **download** function. This downloads the target data set or file with supplied code pages. For Unix files the download function automatically detects the code page to download as IBM-1047 or binary if no code pages are supplied.

Invoking the function

```
window.parent.zosmf.api.download(contentParam);
```

Where **contentParam** is an object with keys of:

Table 606. Parameters of Download function		
Parameter name	Description	Required value
name	Name or full path of the file or data set	Y
member	The name of the data set member	N
volser	The volume serial of the data set to download	N
ibmDataType	The source code page of the file or data set to perform data conversion. Default is IBM-1047	N
contentType	The target code page of the file or data set to perform data conversion. Default is ISO8859-1	N

The function launches the download of the target resource in the web browser.

Return values

N/A. The **download** is a void function.

Example

```
var contentParam = {name: '/var/zosmf/data/logs/zosmfServer/logs/messages.log'};  
window.parent.zosmf.api.download(contentParam);
```

Upload function

If your application allows users to upload a local file to z/OS as a sequential data set, a data set member of a Unix file, consider calling the **upload** function. This creates a new data set or file, which replaces the existing one if the same object exists.

Invoking the function

```
window.parent.zosmf.api.upload(contentParam);
```

Where **contentParam** is an object with keys of.

Table 607. Parameters of the Upload function		
Parameter Name	Description	Required value
callback	The call back function that triggers after the upload action is complete.	Y
type	The resource that the local file is uploaded as. The value should be either 'dataset', 'member', or 'file'. 'dataset' is the default value that is created as a sequential data set.	N
datasetName	The data set name if the local file is uploaded as a data set member.	Y if type = member
unixFilePath	The full path of the Unix directory if the local file is uploaded as a Unix file.	Y if type = file
errorback	The call back function to trigger after the upload action fails.	N
volser	The volume serial of the uploaded data set.	N
useDefaultContentType	Boolean value Y indicates that the default code pages are used for data conversion. N indicates that a dialog prompts the user to select a source and target code pages.	N

Table 607. Parameters of the Upload function (continued)

Parameter Name	Description	Required value
ibmDataType	The target code page of the file or data set to perform data conversion. Default is IBM-1047. If ibmDataType or contentType presents, no code page selection dialog is prompted.	N
contentType	The source code page of the file or data set to perform data conversion. Default is ISO8859-1. If ibmDataType or contentType presents, no code page selection dialog is prompted.	N

The function launches the OS file selector for the user to select a local file to upload as the specified **type**. The function returns to the caller's call back function after the upload completes or fails.

Return values

The **upload** function returns the following values to the call back function:

- **file**
The local file instance.
- **targetName**
 - The full name of data set if the local file is uploaded as a sequential data set. For example,; ZOSMF.SERVER.CERT.
 - The full name of data set with member name if the local file is uploaded as a data set member. For example,; ZOSMF.SERVER.PARMLIB(IZUPRM1).
 - The Unix file path if the local file is uploaded as a file. For example,; /global/zosmf/configuration/local_override.cfg
- **targetVolser**
The target volume if the data set is uploaded with a volser parameter.

Example

```
var param = {} as any;
param.type = 'member';
param.datasetName = 'ZOSMF.PARMLIB.EXP1';
param.callback = (file, targetName, targetVolser) => { };
param.errorback = (file, targetName, error, targetVolser) => { };
window.parent.zosmf.api.upload(param);
```

Functions with z/OSMF window

z/OSMF provides the `zosmfExternalTools` JavaScript API. z/OSMF tasks can use this API to define actions before z/OSMF closes the task, obtain a unique session identifier, and store and manage public objects in z/OSMF core.

Table 608. Function usage with z/OSMF window

Function	Description	Function usage where described
<code>programmaticallyCloseTab</code>	Call this function to request that z/OSMF core close your task.	“programmaticallyCloseTab function” on page 1235
<code>cleanupBeforeDestroyComplete</code>	Call this function to inform z/OSMF core that the actions performed by the <code>cleanupBeforeDestroy</code> function are complete. After which, z/OSMF closes your task.	“cleanupBeforeDestroyComplete function” on page 1236
<code>getUserSessionId</code>	Call this function to retrieve the unique session identifier that z/OSMF core creates for each user and the z/OSMF instance combination.	“getUserSessionId function” on page 1236

Table 609 on page 1234 lists the functions that you can define and dynamically add to the `zosmfExternalTools` API. z/OSMF core calls these functions when specific events are triggered.

Table 609. Functions you can add to the `zosmfExternalTools` API

Function	Usage	When called	Where it is described
isContentChanged	Define this function if you want your task to check for unsaved changes before z/OSMF closes the task.	z/OSMF core calls your <code>isContentChanged</code> function when: <ul style="list-style-type: none">• A user clicks the X icon to close the z/OSMF tab that contains your task.• Your task calls the <code>programmaticallyCloseTask</code> function.	“isContentChanged function” on page 1237
shouldClose	Define this function if you want to delay the close task request for an unspecified period of time so that your task can perform some additional actions.	z/OSMF core calls your <code>shouldClose</code> function when a user clicks the X icon to close the z/OSMF tab that contains your task.	“shouldClose function” on page 1238

Table 609. Functions you can add to the `zosmfExternalTools` API (continued)

Function	Usage	When called	Where it is described
cleanupBeforeDestroy	Define this function if you want to delay the close task request for up to 1 second so that your task can perform some cleanup actions before z/OSMF closes the task.	z/OSMF core calls your <code>cleanupBeforeDestroy</code> function when a user: <ul style="list-style-type: none"> • Logs out of z/OSMF. • Click the X icon to close the z/OSMF tab that contains your task. • Closes the browser tab or window. • Changes the URL in the browser and redirects away from z/OSMF. 	“cleanupBeforeDestroy function” on page 1239

***programmaticallyCloseTab* function**

If your task overrode or intercepted a user's request to close your task, call the **programmaticallyCloseTab** function to request z/OSMF core to close your task. Otherwise, your task remains open.

Invoking the function

To call the **programmaticallyCloseTab** function, use the syntax that is shown in [Figure 514 on page 1235](#).

```
window.parent.programmaticallyCloseTab(pluginId, taskId);
```

Figure 514. Syntax to use to call the *programmaticallyCloseTab* function

where,

pluginId

Unique identifier that is assigned to the plug-in that contains the task.

taskId

Unique identifier that is assigned to the task that you want to close.

When your task calls this function, z/OSMF core looks up the *pluginId* and *taskId* and attempts to close the corresponding z/OSMF tab. This close task request has the same behavior as the close request that is submitted when a user clicks the X to close the task tab.

For this service to work, z/OSMF core must be the parent of the tab that contains your task. For example, z/OSMF can use the **programmaticallyCloseTab** function to close your task when your task is launched from the z/OSMF navigation tree or launched with context by the Application Linking Manager.

z/OSMF cannot use this service to close your task if the task is open in another browser tab or window. Likewise, you cannot use this service or if you made your task a link and launched the task directly.

Example

Suppose you have a plug-in with the ID *com.company.product*, and it contains a task with the ID *Product Name*. To close the task that uses the **programmaticallyCloseTab** function, you can use the sample code that is provided in [Figure 515 on page 1236](#).

```

window.zosmfExternalTools.isContentChanged = () => {
    return false;
}
//Call the parent to refer to z/OSMF core. Then, call the function.
window.parent.programmaticallyCloseTab("com.company.product","Product Name");
}

```

Figure 515. Sample code for the programmaticallyCloseTab function

***getUserSessionId* function**

z/OSMF core creates a unique ID for the authenticated user for every browser tab or window in which the user has z/OSMF or a z/OSMF task opened. Your task can use the ID for any purpose. To retrieve the ID, call the *getUserSessionId* function.

Invoking the function

To call the *getUserSessionId* function, use the syntax that is shown in [Figure 516 on page 1236](#).

```

window.parent.getUserSessionId();

```

Figure 516. Syntax to use to call the getUserSessionId function

The session ID has the form *userName_timestamp*, where *userName* is the ID the user used to log in to z/OSMF, and the *timestamp* is the date and time that the user logged in to z/OSMF or opened z/OSMF or a z/OSMF task in a new browser tab or window. The timestamp is based on the locale and time zone setting for the user's browser.

Example

```

var userSessionID;
init(){
    //Create a unique session ID for the session to track if multiple
    //instances of the application are open.
    userSessionID=window.parent.getUserSessionId();
    //This session ID can be used in any client side logging.
}

```

Figure 517. Sample code for the getUserSessionId function

***cleanupBeforeDestroyComplete* function**

If your task defined a **cleanupBeforeDestroy** function, call the **cleanupBeforeDestroy** function to inform z/OSMF core that your cleanup actions are complete, and that your task is ready to be closed.

Invoking the function

To call the **cleanupBeforeDestroy** function, use the syntax that is shown in [Figure 518 on page 1236](#).

```

window.parent.zosmfTools.cleanupBeforeDestroyComplete(obj);

```

Figure 518. Syntax to use to call the cleanupBeforeDestroyComplete function

Where,

obj

Name of the parameter that was passed to the **cleanupBeforeDestroy** function.

The **cleanupBeforeDestroy** function must be called after your **cleanupBeforeDestroy** function completes. Otherwise, z/OSMF might close the task before the cleanup actions are complete.

For example, if your **cleanupBeforeDestroy** function is performing an XMLHttpRequest (XHR) call, the **cleanupBeforeDestroy** function must be called at the end of the XHR call and not immediately after the XHR call is issued.

Note: If your task does not call the **cleanupBeforeDestroy** function within 1 second after z/OSMF invokes your **cleanupBeforeDestroy** function or if errors occur with the **cleanupBeforeDestroy** function, z/OSMF will close your task after 1 second elapses.

Example

```
window.zosmfExternalTools.cleanupBeforeDestroy = (obj) => {  
  //A synchronous cleanup method that cleans up objects on the client  
  //to free up memory.  
  myApi.cleanupMethod();  
  //Inform z/OSMF core that the cleanup actions are complete  
  //and that the task is ready to be closed.  
  window.parent.zosmfTools.cleanupBeforeDestroyComplete(obj);  
};
```

Figure 519. Sample code for the `cleanupBeforeDestroyComplete` function

If `myApi.cleanupMethod()` is an asynchronous method, call the **cleanupBeforeDestroy** function after the asynchronous action completes. Otherwise, the asynchronous method might not have time to complete.

There are also functions you can add dynamically to customize the window actions. You must define the **zosmfExternalTools** object in the window object of your application. [Figure 520 on page 1237](#) provides sample code.

```
window.zosmfExternalTools = {};
```

Figure 520. Sample JavaScript code for instantiating the `zosmfExternalTools` object

isContentChanged function

If your task allows users to save input values, modified content, or selections, consider defining an **isContentChanged** function that checks for unsaved changes before z/OSMF closes the task. This prevents users from inadvertently discarding their work.

Invoking the function

The **isContentChanged** function must be defined off the `zosmfExternalTools` object. To define the **isContentChanged** function, use the syntax that is shown in [Figure 521 on page 1237](#).

```
window.zosmfExternalTools.isContentChanged = ( ) => {  
  //Define your function here.  
  //Return either true or false.  
  return true|false;  
}
```

Figure 521. Syntax to use for the `isContentChanged` function

z/OSMF core calls the **isContentChanged** function that is provided for your task when a user clicks the X icon to close the tab that contains your task or when your task calls the `programmaticallyCloseTask` function.

Return values

The **isContentChanged** function returns a Boolean value, which indicates the following:

true

Indicates that changes are pending. In this case, z/OSMF core displays a prompt that warns the user that unsaved changes are to be discarded. It gives the user the option to proceed with closing the task or to cancel the close task request. [Figure 522 on page 1238](#) depicts a sample prompt.

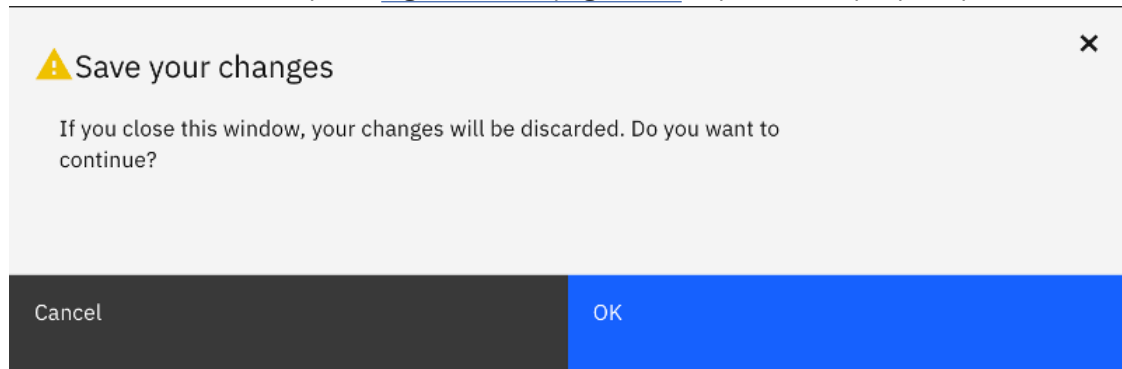


Figure 522. Sample confirmation window for a close task request

False

Indicates that no changes are pending. In this case, z/OSMF core closes the task tab and does not display a prompt.

Note: If the `zosmfExternalTools` object or the **isContentChanged** functions do not exist, z/OSMF core displays a prompt, regardless of whether changes are pending.

Example

Suppose your plug-in contains a task that has multiple editable fields on a single page. When the user clicks the X icon to close your task tab, you want to check for changes for each field. [Figure 523 on page 1238](#) provides sample code that you can use for this scenario.

```
//When the page is created, call the internal _isContentChanged() function.
ngAfterViewInit() {
  window.zosmfExternalTools = {};

  //Define the isContentChanged function.
  window.zosmfExternalTools.isContentChanged = () => {
    return this._isContentChanged();
  }
}
//Determine if the content was changed.
_isContentChanged(){
  ... },
```

Figure 523. Sample code for the isContentChanged function

shouldClose function

When a close task request is submitted, if your task needs to perform cleanup actions or collect information from users before z/OSMF closes the task, consider defining a **shouldClose** function to override requests to close your task. Doing so gives your task an unlimited amount of time to perform the actions, and allows your task to inform z/OSMF when it is ready to be closed.

Invoking the function

z/OSMF core calls the **shouldClose** function that is provided for your task when a user clicks the X icon to close the tab that contains your task. The **shouldClose** function must be defined off the

zosmfExternalTools object. To define the **shouldClose** function, use the syntax that is shown in [Figure 524 on page 1239](#).

```
window.zosmfExternalTools.shouldClose = ( ) => {  
  //Define your function here.  
  //Return either true or false.  
  return true|false;  
}
```

Figure 524. Syntax to use for the shouldClose function

Return values

The **shouldClose** function returns a Boolean value, which indicates the following:

true

Indicates that z/OSMF core should close the task.

false

Indicates that z/OSMF core should not close the task at this time. The task submits a close task request when it is ready to be closed. Return **false** if you need to perform additional steps or cleanup before you close the task. The task remains open until your task calls the **programmaticallyCloseTab** function.

Tip: There is no time limit on how long the tab can remain open after a user requests for it to be closed. If the additional actions take more than a few seconds to complete, consider providing an indicator so that users know the close request is being processed.

Note: If the zosmfExternalTools object or the **shouldClose** functions do not exist, z/OSMF core closes the task.

Example

```
var functionToCall = ( ) => {  
  //Task tab is now open indefinitely.  
  //Perform all cleanup work.  
  ....  
  //When its finished, call the programmaticallyCloseTab function.  
  window.parent.programmaticallyCloseTab(myPluginId,myTaskId);  
}  
window.zosmfExternalTools.shouldClose = ( ) => {  
  if(longCleanupNeeded){  
    //If cleanup actions will take longer than one second,  
    //call the shouldClose function, and sSet the cleanup  
    //function to start asynchronously after the shouldClose  
    //function returns false.  
    setTimeout(functionToCall,5);  
    return false;  
  }else{  
    //if we dont need more time and should close  
    return true;  
  }  
}
```

cleanupBeforeDestroy function

When a close task request is submitted, if your task needs to perform cleanup actions before z/OSMF closes the task, such as ending a TSO/E address space, consider defining a **cleanupBeforeDestroy** function. Doing so delays a close task request for up to 1 second so that your task can perform the cleanup actions.

Invoking the function

The **cleanupBeforeDestroy** function must be defined off the zosmfExternalTools object. To define the **cleanupBeforeDestroy** function, use the syntax that is shown in [Figure 525 on page 1240](#).

```

init(){
  window.zosmfExternalTools = new zosmfExternalTools();
  //Define the cleanupBeforeDestroy function.
  window.zosmfExternalTools.cleanupBeforeDestroy = (obj) => {
    //Perform some cleanup
    .....
    //When cleanup is complete, call the cleanupBeforeDestroyComplete
    //function using the same parameter you used for the
    //cleanupBeforeDestroy function.
    window.parent.zosmfTools.cleanupBeforeDestroyComplete(obj);
  } }

```

Figure 525. Syntax to use for the `cleanupBeforeDestroy` function

Where,

obj

Object that z/OSMF core creates to identify the task. You can specify any parameter name, but you must use the same name for the **cleanupBeforeDestroy** and **cleanupBeforeDestroyComplete** functions.

z/OSMF core calls the **cleanupBeforeDestroy** function that is provided for your task when a user:

- Logs out of z/OSMF.
- Click the X icon to close the z/OSMF tab that contains your task.
- Closes the browser tab or window.
- Changes the URL in the browser and redirects away from z/OSMF.

z/OSMF core calls the **cleanupBeforeDestroy** function every time one of the previously mentioned events occurs. If a user has multiple instances of your task open or is working with multiple tasks in a plug-in, ensure that you do not release resources until the last task, or the last instance of your task, is closed.

After z/OSMF core calls your **cleanupBeforeDestroy** function, the task will remain open until you call the **cleanupBeforeDestroyComplete** function or until 1 second elapses, at which point, z/OSMF core will automatically close your task.

Tip: If you define a **cleanupBeforeDestroy** function, you do not need to define a **shouldClose** function because z/OSMF assumes **shouldClose** returns **false**.

Example

```

window.zosmfExternalTools.cleanupBeforeDestroy = function(obj) {
  //A synchronous cleanup method that cleans up objects on the client
  //to free up memory.
  myApi.cleanupMethod();
  //Inform z/OSMF core that the cleanup actions are complete
  //and that the task is ready to be closed.
  window.parent.zosmfTools.cleanupBeforeDestroyComplete(obj);
};

```

Figure 526. Sample code for the `cleanupBeforeDestroy` function

If `myApi.cleanupMethod()` is an asynchronous method, call the **cleanupBeforeDestroyComplete** function after the asynchronous action completes. Otherwise, the asynchronous method might not have time to complete.

Using the Application Linking Manager JavaScript APIs

If your installation uses multiple, disparate web interfaces to manage your z/OS systems, use the z/OSMF Application Linking Manager to connect the applications. Doing so allows one task or application, an event requester to request that specific function or context to be launched in another task or application, the event handler, providing a smoother transition between applications.

z/OSMF provides the following resources for working with the Application Linking Manager:

- **Application Linking Manager task**, which provides a graphical user interface that you can use to add, query, or remove event type and event handler definitions.
- **Application Linking Manager REST APIs**, which are a set of REST services that allows a client application to add, query, or remove event type and event handler definitions.
- **AppLinker JavaScript API**, which is a set of JavaScript services that allows a client application to send events to the Application Linking Manager or to define the context to be displayed. The JavaScript services are applicable only if you are creating your own z/OSMF plug-in.

The remainder of this section describes the AppLinker JavaScript API. For information about the Application Linking Manager task, see the z/OSMF online help. For details about the REST APIs, see [“Application Linking Manager interface services” on page 8](#).

z/OSMF predefines several event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF” on page 10](#).

Retrieving the AppLinker instance

To participate in the application linking process as an event requestor or an event handler, your task needs access to the functions provided in the AppLinker API, which is located at the following path: /zosmf/js/zosmf/izual.

To access the functions in the API, you must import and instantiate the AppLinker API in your task’s HTML file. Sample code is provided in [Figure 527 on page 1241](#).

```
var applinker = window.parent.zosmf.api.pluginAppLinker
```

Figure 527. Sample code for importing and instantiating the AppLinker API

Functions provided in the AppLinker API

[Table 610 on page 1241](#) lists the functions that are provided in the AppLinker API.

<i>Table 610. Functions provided in the AppLinker API</i>		
Function	Usage	Where described
sendEvent	If your task is an event requestor, call this function to send an event to the Application Linking Manager.	“sendEvent function” on page 1243
getHandlers	If your task is an event requestor, call this function to determine whether handlers are available to process an event.	“getHandlers function” on page 1244
hasLaunchContext	If your task is an event handler and supports the <i>launch with context</i> or <i>launch with context and reload</i> launching option, call this function to determine if your task is being loaded as a result of an application linking event.	“hasLaunchContext function” on page 1245
getEventFromUrl	If your task is an event handler and the hasLaunchContext function returns <i>true</i> , call the getEventFromUrl function to retrieve the event information that was supplied with the event.	“getEventFromUrl function” on page 1246

Table 610. Functions provided in the Applinker API (continued)

Function	Usage	Where described
subscribe	If your task is an event handler and supports the <i>launch with context and switch</i> launching option, call this function to define a JavaScript function that z/OSMF core will call when an event of the specified type is delivered to your task.	“subscribe function” on page 1248
onLoadingComplete	If your task is an event handler and supports the <i>launch with context and switch</i> launching option, after your task subscribes to the event types it can handle, call this function to inform the Application Linking Manager that your task is ready to handle events.	“onLoadingComplete function” on page 1250

Using the Applinker functions in the application linking process

The application linking process consists of the following steps:

1. An event requestor defines a user interface control that invokes the *sendEvent* function when a user performs an action.
2. An event requestor calls the *getHandlers* function to determine if handlers are available to process the request. If handlers are not available, the event requestor might perform an action such as disabling or hiding the user interface control.
3. A user performs an action on the user interface control that triggers the call of the *sendEvent* function.
4. The Application Linking Manager searches the set of known event types for the type that is identified by the event.
5. If a match is found, the Application Linking Manager searches for event handlers that are registered for this event type. If only one handler is found, it is launched. Otherwise, the user is prompted to select the handler to launch.
6. After the handler is identified, z/OSMF core identifies the launch context the handler supports.
7. If the launch context is *launch without context*, z/OSMF core uses the URL provided in the handler definition to launch the handler. If the handler is already open, it receives focus.
8. If the launch context is *launch with context*, *launch with context and reload*, or *launch with context and switch*, z/OSMF core does the following:
 - Appends the event type and parameters to the URL provided in the handler definition.
 - Completes one of the following actions:
 - If the launch context is *launch with context*, z/OSMF core uses the modified URL to launch the handler. If the handler is already open, it receives focus.
 - If the launch context is *launch with context and reload*, z/OSMF core uses the modified URL to launch the handler. If the handler is already open, a message is displayed warning the user that the current context will be overwritten.
 - If the launch context is *launch with context and switch*, the following steps are completed:
 - a. z/OSMF core uses the modified URL to launch the handler.
 - b. If the handler is loading for the first time, the handler calls the *subscribe* function for each event type to which it wants to subscribe and provides the function that z/OSMF core calls to determine the context to display.
 - c. After the handler subscribes to all the event types it can process, the handler calls the *onLoadingComplete* function to inform z/OSMF core that it is ready to accept events.

- d. When an event occurs, z/OSMF core verifies that the handler has called the *onLoadingComplete* function. If the function has been called, z/OSMF core searches the list of handlers that have subscribed for this event type, and identifies the callback function for the selected handler.

Important: Register your task for each event type to which your task subscribes. Otherwise, z/OSMF core will not use the subscription because users will not have the option of selecting your task as the handler for the event.

- e. z/OSMF core calls the *subscribe* callback function.
 - f. The function returns the context for z/OSMF core to display.
 - g. z/OSMF core displays the context and finishes loading the handler.
9. To display the correct context for the *launch with context* and *launch with context and reload* launching options, the handler must do the following while it is being loaded:
 - a. Call the *hasLaunchContext* function to determine whether an event occurs.
 - b. Call the *getEventFromUrl* function to extract the event information from the URL that z/OSMF used to launch the handler.
 - c. Use the event information to display the correct context. For example, the handler might call another JavaScript function, which you provide, that can process the event information and return the correct context.

sendEvent function

If your task is an event requestor, call the *sendEvent* function to send an event to the Application Linking Manager. The *sendEvent* function initiates the application linking process.

Overview

The *sendEvent* function is typically attached to a user interface control, such as a link, a button, or an action. When the user interacts with this control, the task calls the *sendEvent* function, which supplies the event type ID and parameters to the Application Linking Manager. The Application Linking Manager performs several actions to identify the handler to be launched, and then it launches the handler.

To complete the application linking process, the event type specified for the *sendEvent* must be registered with the Application Linking Manager, and at least one handler must be available to process the request.

To minimize errors during the application linking process, call the *getHandlers* function and verify that at least one handler is available to handle the event. If no handlers are available, consider hiding or disabling the user interface control that calls the *sendEvent* function.

Invoking the function

To call the *sendEvent* function, use the syntax shown in [Figure 528 on page 1243](#).

```
window.parent.zosmf.api.pluginAppLinker.sendEvent(eventTypeId, params);
```

Figure 528. Syntax to use to call the *sendEvent* function

where,

eventTypeId

ID that identifies the type of event.

params

JSON object array that contains the name and value for each parameter that your task will provide with the event. Specifying parameters is optional. The syntax to use follows:

```
{parm1: value1, parm2: value2, parm3: value3}
```

Example

```
window.parent.zosmf.api.pluginAppLinker.sendEvent("IBM.ZOSMF.VIEW_DATASET",{dataSetName:"myDataSet"});
```

Figure 529. Sample code for the sendEvent function

getHandlers function

If your task is an event requestor, consider calling the *getHandlers* function to determine if handlers are available to process your request. The *getHandlers* function does not initiate application linking. It helps your task determine if application linking is possible.

Overview

The *getHandlers* function identifies handlers that satisfy the following criteria:

- The handler is registered as a handler for the event type.
- The handler is enabled for the event type.
- The user is authorized to access the handler.

If one or more handlers satisfy this criteria, application linking is possible. Otherwise, application linking is not possible. In the latter case, consider hiding or disabling the user interface control that will initiate the application linking process. Doing so increases the usability of your task because the control is enabled or displayed only when the Application Linking Manager can successfully process the user's request.

Tip: If the *getHandlers* function does not find handlers that satisfy the aforementioned criteria, ensure that the event type is registered with the Application Linking Manager. If the event type is registered, verify that the event type ID is spelled correctly in your task.

Invoking the function

To call the *getHandlers* function, use the syntax shown in [Figure 530 on page 1244](#).

```
//Define a function for the getHandlers function to call if the request
//completes without errors.
var callback = function(response){
    //Specify what to do if there are handlers.
}
else {
    //Specify what to do if there are no handlers.
}
}

//Define a function for the getHandlers function to call if errors
//occur with the request.
var errback = function(error){
    //Specify how to proceed.
}

//Call the getHandlers function.
window.parent.zosmf.api.pluginAppLinker.getHandlers(eventTypeId,callback,errback);
```

Figure 530. Syntax to use to call the getHandlers function

where,

response

JSON object array, which is provided by the *getHandlers* function, that contains the name of each handler that is available to process the event. To access the handlers in the array, use `response.results`.

errors

JSON object array, which is provided by the *getHandlers* function, that contains the error messages the function received. To access the messages in the array, use `error.messages`.

eventId

ID that identifies the type of event.

callback

Function, which you provide, that the *getHandlers* function will call if it completes without errors.

errback

Function, which you provide, that the *getHandlers* function will call if errors occur when it is processing the request.

Example

```
var callback = function(response){
    if(response && response.results && response.results.length>0){
        //Handlers exist; enable the user controls.
    }else{
        //No handlers exist; disable the user controls.
    }
}
var errback = function(error){
    //Error occurred when retrieving handlers; retrieve the messages.
    var messages = error.messages;
}
window.parent.zosmf.api.pluginApplinker.getHandlers("IBM.ZOSMF.VIEW_DATASET",callback,errback);
```

hasLaunchContext function

If your task is an event handler, when your task is loading, call the *hasLaunchContext* function to determine if your task is being launched as a result of an application linking event. Doing so allows your task to determine if it needs to collect event information and display a specific context, or if it can display the main page.

Invoking the function

Call the *hasLaunchContext* function only if the launching option specified for your task in the handler definition is *launch with context* or *launch with context and reload*. For the *launch without context* launching option, z/OSMF core does not collect event information because there is no context to display; therefore, it is not necessary for your task to call the *hasLaunchContext* function.

For the *launch with context and switch* launching option, z/OSMF core alerts your task that an event has occurred when it calls the callback function you specified for the subscribe function; therefore, it is not necessary for your task to call the *hasLaunchContext* function to determine if an event has occurred.

To call the *hasLaunchContext* function, use the syntax shown in [Figure 531 on page 1245](#).

```
window.parent.zosmf.api.pluginApplinker.hasLaunchContext();
```

*Figure 531. Syntax to use to call the *hasLaunchContext* function*

Return values

The *hasLaunchContext* function returns a Boolean value, which indicates the following:

true

Indicates that the Application Linking Manager delivered an event to the task. Call the *getEventFromUrl* function to parse the URL that z/OSMF core used to launch the task and retrieve the event information.

false

Indicates that the Application Linking Manager has not delivered an event to the task. In this case, z/OSMF core will launch the task without context.

Example

```
function init(){
{
  if(window.parent.zosmf.api.pluginApplinker.hasLaunchContext()){
    //The task was opened with application linking; therefore, update context.

    //Use the getEventFromURL function to obtain the context.
    var result = window.parent.zosmf.api.pluginApplinker.getEventFromURL();
    var eventType = result.type;
    var params = result.params;

    //Use the eventType and parameters to switch the context.
  }
  else{
    //The task was not opened with application linking; therefore, display
    //the standard starting page.
  }
}
}
```

Figure 532. Sample code for the *hasLaunchContext* function

getEventFromUrl function

If your task is an event handler and the *hasLaunchContext* function returns *true*, call the *getEventFromUrl* function to parse the URL that z/OSMF core used to launch your task. Doing so will provide your task with the event type ID and parameters it needs to display the correct context. Otherwise, your task will launch without context.

Invoking the function

To call the *getEventFromUrl* function, use the syntax shown in [Figure 533 on page 1246](#).

```
window.parent.zosmf.api.pluginApplinker.getEventFromUrl();
```

Figure 533. Syntax to use to call the *getEventFromUrl* function

Return values

The *getEventFromUrl* function returns an event object that includes the following information:

type

The event type ID supplied with the event.

params

Array that contains the name and value of each parameter supplied with the event.

Tip: While your task is loading, it must use the event information to display the correct context. For example, your task might call another JavaScript function, which you provide, that can process the event information and return the correct context.

Expected results

The event information returned by the *getEventFromUrl* function depends on the launching option that is specified in the event handler definition for your task and whether your task is already open when an event is received.

Table 611 on page 1247 describes the expected result for each launching option and task state (open or closed) combination.

Table 611. Expected results by launching option and task state	
Launching option and task state	Expected result
The launching option is <i>launch without context</i> and the task is closed.	<p>The <i>getEventFromUrl</i> function will return <i>null</i>, and z/OSMF core will launch the task using the URL supplied in the handler definition.</p> <p>Tip: z/OSMF core does not collect event information for this launching option because there is no context to display; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function.</p>
The launching option is <i>launch without context</i> and the task is already open.	<p>The <i>getEventFromUrl</i> function will return <i>null</i>, and z/OSMF core will bring the existing task tab into focus.</p> <p>Tip: z/OSMF core does not collect event information for this launching option because there is no context to display; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function.</p>
The launching option is <i>launch with context</i> and the task is closed.	<p>The <i>getEventFromUrl</i> function will return the event type ID and parameters for the current event, and the task will display the current context.</p>
The launching option is <i>launch with context</i> and the task is already open.	<p>The <i>getEventFromUrl</i> function will return one of the following values:</p> <ul style="list-style-type: none">• The event type ID and parameters for the event that z/OSMF core used to initially open the task.• <i>Null</i> if the task was not initially opened as a result of an application linking event. <p>z/OSMF core will bring the existing task tab into focus, and the task will display its last context.</p>
The launching option is <i>launch with context and reload</i> and the task is closed.	<p>The <i>getEventFromUrl</i> function will return the event type ID and parameters for the current event, and the task will display the current context.</p>
The launching option is <i>launch with context and reload</i> and the task is already open.	<p>z/OSMF core brings the existing tab into focus and displays a message, which warns the user that the current context will be overwritten and gives the user the option to proceed with displaying the new context or keeping the previous context.</p> <p>If the user clicks OK, the <i>getEventFromUrl</i> function will return the event type ID and parameters for the new event, and the task will display the new context.</p> <p>If the user clicks Cancel, the <i>getEventFromUrl</i> function will return one of the following values, and the task will display its last context:</p> <ul style="list-style-type: none">• The event type ID and parameters for the previous event.• <i>Null</i> if no event has occurred.

Table 611. Expected results by launching option and task state (continued)

Launching option and task state	Expected result
The launching option is <i>launch with context and switch</i> and the task either is closed or is already open.	<p>The <i>getEventFromUrl</i> function will return the event information for the current event, and your task will determine the context to display based on the callback function you provided as a parameter for the <i>subscribe</i> function.</p> <p>Tip: z/OSMF core supplies your task with the event type ID and event parameters when it calls the <i>subscribe</i> callback function; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function to retrieve the event information.</p> <p>For information about the <i>subscribe</i> function, see “subscribe function” on page 1248.</p>

Example

```
function init(){
{
  if(window.parent.zosmf.api.pluginAppLinker.hasLaunchContext()){
    //The task was opened with application linking; therefore, update context.

    //Use the getEventFromURL function to obtain the context.
    var result = window.parent.zosmf.api.pluginAppLinker.getEventFromURL();
    var eventType = result.type;
    var params = result.params;

    //Use the eventType and parameters to switch the context.

  }
  else{
    //The task was not opened with application linking; therefore, display
    //the standard starting page.
  }
}
```

Figure 534. Sample code for the *getEventFromUrl* function

subscribe function

If your task is an event handler and supports the *launch with context and switch* launching option, you must define the context for your task to display. To do so, your task must call the *subscribe* function for each event type it can handle and provide a JavaScript function for z/OSMF core to call when the Application Linking Manager delivers an event of the specified type.

Difference between subscribing and registering for an event type

Subscribing your task to an event type is different than registering your task as a handler for an event type. Registration is done regardless of launch context and it informs z/OSMF core that your task can handle events of the specified type. As such, z/OSMF core will present your task to users as a possible handler for events of that type.

Subscriptions are supported for the *launch with context and switch* launching option only, and it tells z/OSMF core how to process events of the specified type.

For the *launch with context and switch* launching option, your task must register for and subscribe to an event type. If your task subscribes but does not register as a handler for an event type, z/OSMF core will not use your task's subscription because it is impossible for users to select your task as the handler for the event type. If your task registers but does not subscribe to an event type, z/OSMF core will launch your task without context.

Invoking the function

Your task must call the *subscribe* function while it is still loading so that when it is loaded the correct context is shown. To call the *subscribe* function, use the syntax shown in [Figure 535 on page 1249](#).

```
//Define a function that specifies how to handle the event.
var subscribeCallback = function(params, eventData) {
    //Define your callback function here.
}

//Call the subscribe function.
window.parent.zosmf.api.pluginAppLinker.subscribe(eventTypeId, subscribeCallback);
```

Figure 535. Syntax to use to call the subscribe function

where,

params

Object that contains an array of the parameters supplied with the event. z/OSMF core will provide this object when it calls your callback function.

eventData

Object that contains the event type ID. z/OSMF core will provide this object when it calls your callback function.

eventTypeId

ID of the event type to which your task is subscribing.

subscribeCallback

Function, which you provide, that specifies the context for your task to display when z/OSMF core delivers an event with the specified ID to your task.

Tip: If your *subscribeCallback* function is a function in your current object, you might be required to use `dojo.hitch(this,"subscribeCallback")` to specify it in the *subscribe* function call.

To subscribe to multiple event types, your task must call the *subscribe* function for each event type and, more than likely, provide a unique subscribe callback function for each call. For example, if your task is subscribing to three event types, it will call the *subscribe* function three times and, possibly, provide three unique callback functions.

After your task subscribes to all the event types it can handle, your task must call the *onLoadingComplete* function to inform z/OSMF core that it is ready to accept events. If your task does not call the *onLoadingComplete* function, z/OSMF core will not deliver the event that initially launched your task. z/OSMF core will, however, deliver subsequent events if your task has subscribed to the corresponding event type.

When z/OSMF core delivers an event to your task, z/OSMF core searches the list of tasks that have subscribed for the corresponding event type and identifies the callback function for your task. Then, z/OSMF core calls the callback function, and displays the context the function returns.

Tip: Your task does not have to unsubscribe the callback functions because z/OSMF core automatically unsubscribes the callback functions when your task uninitializes the AppLinker API.

Example

```
var viewDataset = function(response){
    //View the parameters in the response, and perform the context switching.
}

function init(){
    window.parent.zosmf.api.pluginAppLinker.subscribe("IZU.ZOSMF.VIEW_DATASET",viewDataset);
}
```

Figure 536. Sample code for the subscribe function

onLoadingComplete function

If your task is an event handler and supports the *launch with context and switch* launching option, after your task subscribes to the event types it can handle, call the *onLoadingComplete* function to inform z/OSMF core that your task is ready to receive events.

Invoking the function

To call the *onLoadingComplete* function, use the syntax shown in [Figure 537 on page 1250](#).

```
window.parent.zosmf.api.pluginAppLinker.onLoadingComplete(suppressInitialEvents);
```

Figure 537. Syntax to use to call the *onLoadingComplete* function

where, *suppressInitialEvents* is a Boolean variable that indicates the following:

true

Indicates that z/OSMF core will suppress the event that was used to initially launch your task. That is, z/OSMF core will ignore the first event, but will deliver subsequent events to the task.

Tip: Consider setting the *suppressInitialEvents* variable to *true* if you always want your task to load a certain way. For example, you might opt to set this variable to *true*, if you want your task to display a warning message or other important information that users must review before proceeding with your task.

If you want your task to process the initial event at a later time, assuming that no subsequent events have been received, you can retrieve the event information by calling the *getEventFromUrl* function and pass that information to the callback function that z/OSMF core would have called.

false

Indicates that z/OSMF core will deliver the event that was used to initially launch your task so that your task can display the correct context.

Tip: If you want to set the *suppressInitialEvents* variable to *false*, you can use the syntax that is provided in either of the following options:

```
//Option 1:  
call window.parent.zosmf.api.pluginAppLinker.onLoadingComplete(false);  
  
//Option 2:  
call window.parent.zosmf.api.pluginAppLinker.onLoadingComplete();
```

Example

```
function init(){  
    setupMyPlugin();  
}  
  
function setupMyPlugin(){  
    //Perform setup actions.  
    ...  
    //Call the onLoadingComplete function to indicate that loading is  
    //complete so the task can switch context.  
    window.parent.zosmf.api.pluginAppLinker.onLoadingComplete(false);  
}
```


Logging client messages in the z/OSMF log

z/OSMF provides a client side logging framework, which you can use to log your plug-in's client messages in the z/OSMF log. Using the client side logger helps simplify debugging activities because messages for all z/OSMF plug-ins are included in the same log.

For more information about the z/OSMF log, including how to access and view the log, see the topic about working with z/OSMF runtime log files in [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Setting up client side logging

To implement client side logging, add code to your plug-in that:

- Specifies the name and location of the Dojo package that contains the client side logger.
- Imports the client side logger package.
- Creates an instance of the client side logger, and assigns a unique identifier to the logger. To uniquely identify your plug-in's client side logger, include the name of your company, product, and module in the identifier, and specify "ui" as the lower level qualifier. For example, `com.ibm.zosmf.NavigationTree.ui`.
- Defines the methods for the logging levels to be included in the z/OSMF log. The logging levels include: FINE, FINER (trace), FINEST (debug), INFO, WARNING, and SEVERE (fatal).

To help you identify your plug-in's messages in the log, append a prefix to each message that contains the module name, package name, and method name.

```
var LOGPREFIX=MODULE+" "+PACKAGE_NAME+" "+methodName+": ";  
LOGGER.entering(LOGPREFIX+"entry stuff");
```

Figure 538. Sample code for creating a log prefix

For sample log data, see “[Sample z/OSMF client side log data](#)” on page 1253.

- Issues JavaScript log statements that match the logging levels you specified. When issued, the client side logger stores the statements in a message queue until the interval, threshold, or trigger is activated, as follows:

interval

By default, the client side logger sends the queued messages to the z/OSMF log every 60 seconds.

threshold

By default, the message queue can contain a maximum of 50 JavaScript log statements. When this threshold is reached, the client side logger sends the queued messages to the z/OSMF log.

trigger

By default, the push level for the client side logger is INFO. When the severity of a JavaScript log statement equals or exceeds the push level, the client side logger sends the queued messages to the z/OSMF log.

Note: To prevent denial of service attacks or filling the log with useless data, the client side logger logs only messages that your plug-in issues on behalf of z/OSMF authenticated users.

Sample code for setting up client side logging

[Figure 539 on page 1252](#) provides sample code that you can use as a reference when setting up the client side logger.

```

var LOGGER = window.parent.LOGGER;
var MODULE = "MYMODULE";
var PACKAGE_NAME = "MYPACKAGE.jsp";

//Methods for logging messages.
init(){
    var methodName="init";
    var LOGPREFIX=MODULE+" "+PACKAGE_NAME+" "+methodName+": ";
    LOGGER.entering(LOGPREFIX+"entry stuff");
    LOGGER.finest(LOGPREFIX+"finest stuff");
    LOGGER.finer(LOGPREFIX+"finer stuff");
    LOGGER.info(LOGPREFIX+"info stuff");
    LOGGER.warning(LOGPREFIX+"warning stuff");
    LOGGER.severe(LOGPREFIX+"severe stuff");
    LOGGER.exiting(LOGPREFIX+"exiting stuff");
}

```

Figure 539. Sample code for setting up client side logging

Methods provided for the client side logger

Table 612 on page 1252 lists the methods you can use when setting up the client side logger, and provides summary descriptions and sample JavaScript code for each method.

Table 612. Methods provided for the client side logger		
Method	Description	Sample Code
log4js.getLogger(logger-ID);	Creates an instance of the client side logger for your plug-in, and assigns a unique identifier (<i>logger-ID</i>) to the logger.	<pre>var LOGGER = window.parent.LOGGER;</pre>
loggerName.log-level(param);	Defines the log levels (<i>log-level</i>) you want the specified logger (<i>loggerName</i>) to include in the z/OSMF log, and takes a single string parameter (<i>param</i>) that specifies the information to log for each message. For the <i>loggerName.severe(param)</i> method, the client side logger automatically flushes messages to the z/OSMF log.	<pre>LOGGER.warning(LOGPREFIX+"Test warning message"); LOGGER.severe(LOGPREFIX+" Exception: " + text);</pre>
loggerName.entering(params);	Logs the entry point of each method, and takes zero or more comma-separated parameters (<i>params</i>).	<pre>LOGGER.entering(LOGPREFIX+"entry stuff");</pre>
loggerName.exiting(params);	Logs the exit point of each method, and takes zero or more comma-separated parameters (<i>params</i>).	<pre>LOGGER.exiting(LOGPREFIX+"exiting stuff");</pre>
loggerName.logp(Level.log-level, params)	Allows you to specify the parameters (<i>params</i>) to include in the z/OSMF log for the specified log level (<i>log-level</i>). This method takes a variable number of comma-separated parameters, and is an alternative to the <i>loggerName.log-level(param)</i> method.	<pre>LOGGER.logp(Level.WARNING, PACKAGE_NAME, MODULE, methodName, msg, " targetid=", targetid); LOGGER.logp(Level.SEVERE, PACKAGE_NAME, MODULE, methodName, "Exception received attempting to authenticate user", userid, exception);</pre>
loggerName.isLoggable(Level.log-level)	Verifies that the specified log level (<i>log-level</i>) is currently being logged.	<pre>if(LOGGER.isLoggable(Level.FINEST)){ LOGGER.finest(LOGPREFIX+ "FTP Host : " + dojo.byId('<%= HADDataAdaptor.FTPHOSTNAME %>').value); }</pre>
loggerName.logMessage("msgID", msgvars);	Logs the message ID as well as the values that were substituted for each parameter. The values are stored as an array.	<pre>var msgvars = ["tkdole", 10, "try again"]; LOGGER.logMessage("IZUG400E", msgvars);</pre>
loggerName.turnOnTracing();	Enables tracing by setting the log level to FINER.	<pre>LOGGER.turnOnTracing();</pre>
loggerName.turnOffTracing();	Disables tracing, and sets the log level to its initial level.	<pre>LOGGER.turnOffTracing();</pre>
loggerName.turnOnPopup();	Appends messages to a popup window. The default log level is FINEST. You can also append messages to a popup window by setting the <i>setpopup</i> property to <i>true</i> . For more details, see "Logging messages to a popup window" on page 1254 .	<pre>LOGGER.turnOnPopup();</pre>
loggerName.turnOffPopup();	Stops appending messages to a popup window (default), and closes the popup window. You can also stop appending messages to a popup window by setting the <i>setpopup</i> property to <i>false</i> . For more details, see "Logging messages to a popup window" on page 1254 .	<pre>LOGGER.turnOffPopup();</pre>

Table 612. Methods provided for the client side logger (continued)		
Method	Description	Sample Code
<code>loggerName.setLogToConsole(true false);</code>	If set to <i>true</i> , the client side logger appends the messages to the browser console, for example, Firebug. If set to <i>false</i> (default), the messages are not appended to the browser console.	<code>LOGGER.setLogToConsole(true);</code> <code>LOGGER.setLogToConsole(false);</code>
<code>loggerName.flush();</code>	Sends the queued messages to the z/OSMF log. You can also use the push property to push queued messages to the z/OSMF log. For more details, see "Pushing messages to the z/OSMF log" on page 1254.	<code>LOGGER.flush();</code>

Sample z/OSMF client side log data

The message entries in the z/OSMF log have a uniform structure, which is depicted in [Figure 540 on page 1253](#).

```
2009-04-29T22:08:09.609Z|00000031|com.ibm.zosmf.util.log.servlet.UILoggerServlet|UILoggerServlet::doPost()
FINER: [2009-04-29T22:07:13.938Z] ENTRY MYMODULE MYPACKAGE.jsp init: entry stuff
[tx0000000000002183:zosmfad@localhost (POST) /zosmf/IzuUICommon/UILoggerServlet?preventCache=1240947046138]
2009-04-29T22:08:09.609Z|00000031|com.ibm.zosmf.util.log.servlet.UILoggerServlet|UILoggerServlet::doPost()
FINER: [2009-04-29T22:07:13.474Z] RETURN MYMODULE MYPACKAGE.jsp init: exiting stuff
[tx0000000000002184:zosmfad@localhost (POST) /zosmf/IzuUICommon/UILoggerServlet?preventCache=1240946722135]
```

Figure 540. Sample z/OSMF client side log data

The first line of a log record contains the following data:

- Date and time the message was added to the log in ISO8601 format, set to UTC timezone. Example: 2009-04-29T22:08:09.609Z.
- Thread ID as an 8 digit hex number. Example: 00000031.
- Class name. Example: com.ibm.zosmf.util.log.servlet.UILoggerServlet.
- Method name. Example: UILoggerServlet::doPost().

The next line of a log record contains the following data:

- Logging level. Possible logging levels include FINE, FINER, FINEST, INFO, WARNING, and SEVERE.
- Date and time the message occurred in ISO8601 format, set to UTC timezone. Example: [2009-04-29T22:07:13.938Z].
- Indicator of the beginning (ENTRY) or end (RETURN) of a routine if the logging level is FINER.
- Log prefix if a prefix was specified for the logging level.
- Message ID and message text. Message IDs that begin with "IZU" are part of the z/OSMF product.

If the log record includes an exception, the exception class and the message text are logged next followed by the traceback information that is embedded in the exception. If the exception has attached causes, each cause is also logged with "+->" indicating the start of an attached cause.

The final line of a log record contains the following data:

- Transaction ID, which is an internal counter that applies to all actions between a specific set and is clear of a context.
- User ID of the user who was logged into z/OSMF when the message was issued.
- Host name of the system where the user logged into z/OSMF.
- Servlet "verb". Examples include (GET) and (POST).
- URL of the request and query string.

Modifying the default settings for the client side logger

z/OSMF core provides default settings for several properties used to manage the client side logger. This section explains how to modify those default settings.

The default settings are described in the following sections:

- [“Displaying the default settings” on page 1254](#)
- [“Pushing messages to the z/OSMF log” on page 1254](#)
- [“Logging messages to a popup window” on page 1254](#)
- [“Redirecting messages that cannot be written to the z/OSMF log” on page 1255](#)
- [“Setting the interval for flushing the queue” on page 1255](#)

Displaying the default settings

To display a list of all the default values for the client side logger properties, use the following HTTP request:

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getprops&loggername={name}
```

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "__OPER=getprops&loggername=name" returns all the default settings for the specified client side logger (*name*).

Pushing messages to the z/OSMF log

The client side logger uses the severity of JavaScript log statements to determine when queued messages are pushed to the z/OSMF log. That is, when the severity of a JavaScript log statement equals or exceeds the push level, the client side logger immediately sends the queued messages to the z/OSMF log. The default push level is INFO. To manage the push level, use the following HTTP requests:

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getLevel&loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=setLevel&level={level}&
loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=push&loggername={name}
```

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "__OPER=getLevel&loggername={name}" returns the current push level for the specified client side logger (*name*).
- "__OPER=setLevel&level={level}&loggername={name}" sets the push level (*level*) for the specified client side logger (*name*), and returns the push level. Valid levels are FINE, FINER, FINEST, INFO, WARNING, and SEVERE.
- "__OPER=push&loggername={name}" pushes the queued messages for the specified client side logger (*name*) to the z/OSMF log.

Logging messages to a popup window

By default, the client side logger routes the JavaScript log statements to the z/OSMF log. You can use the *setpopup* property to route the statements to a popup window in the z/OSMF user interface. To indicate where to display the JavaScript log statements, use the following HTTP requests:

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getpopup&loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=setpopup&loggername={name}
&popup=true|false
```

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "__OPER=getpopup&loggername=name" returns *true* if the specified client side logger (*name*) will append messages to a popup window. Otherwise, *false* is returned.
- "__OPER=setpopup&loggername=name&popup=true|false" indicates whether to append messages to a popup window for the specified client side logger (*name*). Set this property to *true* to append messages to a popup window. Otherwise, set this property to *false*.

If you specify *true* for the *setfailnotify* property, messages will also be displayed in a popup window.

Redirecting messages that cannot be written to the z/OSMF log

If an error occurs that prevents the client's messages from being written to the z/OSMF log, you can use the *setfailnotify* property to indicate whether to redirect those messages to a popup window. This failover action allows for the client data to be retained until the error is resolved. By default, this property is set to *false*.

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=getfailnotify&
loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=setfailnotify&
loggername={name}&failnotify=true|false
```

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "__OPER=getfailnotify&loggername=name" returns *true* if the specified client side logger (*name*) will redirect messages to a popup window when an unexpected error occurs. Otherwise, *false* is returned.
- "__OPER=setfailnotify&loggername=name&failnotify=true|false" indicates whether to redirect messages to a popup window for the specified client side logger (*name*). Set this property to *true* to redirect messages to a popup window when an error occurs. Otherwise, set this property to *false*.

Setting the interval for flushing the queue

The client side logger uses the flush interval to determine how often to flush the message queue and send messages to the z/OSMF log. By default, the logger flushes the message queue every 60 seconds. You can use the *setinterval* property to modify this value.

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=getinterval&
loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=setinterval&
loggername={name}&interval={n}
```

where:

- "https://{host}:{port}" specifies the target system address and port.

- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "__OPER=getinterval&loggername=name" returns the flush interval for the specified client side logger (*name*).
- "__OPER=setinterval&loggername=name&interval=n" sets the flush interval for the specified client side logger (*name*) to the specified number of seconds (*n*). You can specify a minimum of 10 seconds, and a maximum of 3600 seconds.

Retrieving files and resources for your application

z/OSMF provides a file retrieval service that a client application can use to retrieve the files and resources required for the application to display and function properly.

To request files and resources from the z/OSMF file retrieval service, use the following URL format:

```
GET https://{host}:{port}/zosmf/IzuUICommon/externalfiles/{resourcePath}
```

where:

- "https://{host}:{port}" specifies the hostname or IP address and the port of the target system.
- "zosmf/IzuUICommon/externalfiles" identifies the z/OSMF file retrieval service.
- "resourcePath" identifies the file to be retrieved. The path must start with the plug-in context root subdirectory, which is specified in the plug-in's property file. For more details about the plug-in context root, see ["Adding your applications to z/OSMF"](#) on page 1268.

When you issue this HTTP request, the file retrieval service:

- Retrieves the specified file from the UNIX file system.
- Provides the file to z/OSMF core to be displayed.
- Loads the code for the user interface.

Typically, the file retrieval service is used to provide z/OSMF core with the URL to use to launch the application when a user clicks the corresponding task name in the z/OSMF desktop. It can also be used to retrieve additional files and resources as requested by your application.

Example

The following example retrieves the file *myapp.js* for the *myapp* application.

```
GET /zosmf/IzuUICommon/externalfiles/myappcontextroot/myapp.js
Host: 1.56.82.158:80
```

Figure 541. Sample request to retrieve a file

The HTTP response body depends on the type of file to be retrieved. For the previous example, the expected response follows:

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2014 05:39:28 +0000GMT
Content-Type: application/x-javascript

(file content...)
```

Figure 542. Sample response for a request to retrieve a file

Authoring end user assistance

z/OSMF provides a help system that familiarizes users with the interface, teaches users the concepts required to perform the supported tasks, and helps users troubleshoot errors and transition from one step to another. z/OSMF allows you to add documentation to the help system so you can provide end user assistance that enables users to effectively and easily use your application.

Overview of the z/OSMF help system

The z/OSMF help system, which is integrated into the software product, contains user assistance for each page, window, message, and action supported in the z/OSMF interface. Users can access the help system by clicking the help link, help button, or message ID link provided in the interface. When clicked, context-sensitive help is displayed within the z/OSMF help system framework, which is depicted in [Figure 543 on page 1257](#).



Figure 543. Screen capture of the z/OSMF help system

The framework displays the table of contents and the requested help file. The table of contents mirrors the navigation tree provided in the z/OSMF interface. That is, an application (task) and its help content are included in the same category. For example, the user interface and help content for the Incident Log task are contained in the Problem Determination category.

The help system contains the following additional categories:

- **Getting started with z/OSMF.** This section introduces users to the common features of z/OSMF including using tables and wizards, logging in, and navigating.
- **z/OSMF messages.** This section provides a detailed explanation of each z/OSMF message; describes the reason codes (if any) that are listed in each message; and, suggests actions you can perform to resolve the issue.
- **Tools and techniques for troubleshooting.** This section describes the tools and techniques that are available for troubleshooting problems with z/OSMF.

Format of the files in the z/OSMF help system

The help files in the z/OSMF help system are stored as help plug-ins. A *help plug-in* is a folder that contains the following content:

- **doc.zip file:** Contains the help files, which are coded using the XHTML tagging language.
- **toc.xml file:** Provides the table of contents for the plug-in in XML format.
- **index.xml file:** Defines the index entries for the plug-in in XML format.
- **plugin.xml file:** Describes the plug-in to the z/OSMF help system using the XML tagging language.
- **nl folder:** Contains language sub-folders, identified by the 2-character language code, for each language into which the help plug-ins are translated. For example, if the help files are translated into Japanese, the *nl* folder will contain a *ja* sub-folder that contains the Japanese version of the content.

To create help plug-ins and add links to your help files, complete the tasks described in the sections that follow.

Creating help plug-ins

A help plug-in contains the files required to display your application's help files in the z/OSMF help system.

Procedure

1. To create a panel help plug-in, which is the plug-in that contains the help files for each page, window, and action provided in your application, complete the following steps:
 - a) Create a folder for the panel help that has the name *com.company-name.task-name.help.doc*, where *company-name* is your company's name and *task-name* is the task name that will be displayed in the z/OSMF desktop. For example, *com.ibm.incidentlog.help.doc*.
 - b) Store the folder in the UNIX file system, and set 755 permissions for the folder. For all files stored in this folder, set 644 permissions.
 - c) Create the help content, and code it using the XHTML tagging language. Then, combine the files into a single compressed folder named *doc.zip*. For more details, see [“Developing panel help” on page 1259](#).
 - d) Create the table of contents in XML format, and name the file *toc.xml*. For instructions, see [“Creating the table of contents” on page 1263](#).
 - e) Assign the plug-in to the same category you plan to use for your application in the z/OSMF desktop. For instructions, see [“Categorizing help plug-ins” on page 1265](#).
 - f) Use the XML tagging language to create a file named *plugin.xml*, which describes the plug-in to the z/OSMF help system. For instructions, see [“Identifying the contents of help plug-ins” on page 1266](#).
 - g) Store the *doc.zip* folder, *toc.xml* file, *index.xml* file, and *plugin.xml* file in the folder you created.
 - h) In the plug-in folder, create an *nl* sub-folder that contains a folder for each language in which the help content is translated, if any. Store the translated files in the correct language folder, which is identified by a 2-character language code. For a list of language codes, see http://www.loc.gov/standards/iso639-2/php/code_list.php.
2. To create a message help plug-in, which is the plug-in that contains the help files for each message your application issues, complete all of the previous steps with the following exceptions:
 - Store the resulting *doc.zip* folder, *toc.xml* file, *index.xml* file, *plugin.xml* file, and *nl* folder in a folder named *com.company-name.task-name.message.help.doc*.
 - Assign the message help plug-in to the z/OSMF messages category.

For information about creating messages and message help, see [“Developing message help” on page 1261](#).

Results

If you created a panel help plug-in named *com.mycompany.mytask.help.doc* that is translated into Spanish and Japanese, the plug-in will have the following structure:

- com.mycompany.mytask.help.doc
 - doc.zip
 - index.xml
 - plugin.xml
 - toc.xml
 - nl
 - es
 - doc.zip
 - index.xml
 - plugin.xml
 - toc.xml
 - ja
 - doc.zip
 - index.xml
 - plugin.xml
 - toc.xml

Developing panel help

A panel-help plug-in contains a set of topics, which are independent units of information that are meaningful when displayed alone. This topic-based structure allows you to create context-sensitive help for each page and window displayed in your application. Topic-based content can contain task, concept, or reference information.

These information types are described in the following sections:

- [“Task topics” on page 1259](#)
- [“Concept topics” on page 1260](#)
- [“Reference topics” on page 1261](#)

To provide context-sensitive help for your application, use the XHTML tagging language to create the following topics:

- An introductory topic that provides a brief description of the task and highlights its key features.
- A separate topic for each page or window in the user interface that explains the purpose of the panel and describes the elements on the panel including any fields, columns, actions, or buttons.
- A separate topic for each action that provides step-by-step instructions for performing the action.
- A separate topic for any concept or reference information required for users to effectively operate your application.

Task topics

A task topic uses a series of steps to explain how to accomplish a goal. Task topics provide procedures, typically in step-by-step instructions. Some task topics might list choices, as bulleted points rather than steps, or they might describe a single action rather than a sequence of steps. Task topics also provide information about the context (where to perform a task and when), the rationale (why perform the task), prerequisites, and examples.

Supertasks (high-level tasks) are the starting points for most users. Steps in a supertask often link to sub-tasks. Each sub-task is documented as a separate task topic. These tasks can be part of one supertask or many. For example, in database programming, opening a database connection is a task that can be reused in several high-level programming tasks.

A task topic documents what users need to know to successfully complete their work. As you write task information, note what concepts need to be documented, and to what depth, if users are to complete the tasks successfully. Consider the probable skills and experience of users, and write with those characteristics in mind. For example, if users need to configure TCP/IP (a task) and might not know about routers, create and link to a concept topic on routers.

Remember these main points when you write task topics:

- Document the steps that users follow to accomplish their goals.
- Use a verb phrase (gerund) as the heading for a task topic.
- Use an opening paragraph to provide context for and introduce the task.
- If the task has a prerequisite, provide that information or provide a link to that prerequisite before the list of steps.
- Use a numbered list for the steps that users must follow to complete the task.
- Write the steps as brief imperative sentences.
- When the user's context changes, introduce the step with a phrase that establishes that new context (for example, On the Configuration page, ...).
- Write one step for each significant user action.
- If a task has more than nine steps, try to divide it into two or more separate tasks.
- If the task is part of a sequence of tasks, provide a link to the next task.

Concept topics

A concept topic describes a system, solution, product, tool, feature, or background information that users need to complete a task. Users typically read concept material before tackling some large project or starting to use a product or tool. In contrast, users need task or reference topics when they perform a task.

A concept topic describes the scope of the topic and clearly defines what the topic is about. Use minimalist writing techniques to create content that users can quickly understand. Beware of turning a concept topic into a white paper by explaining the whole design philosophy of a product or component. Every concept topic performs at least one of the following functions:

- Introduces a solution, process, product, tool, or feature.
- Provides background information and explains issues that users must know before working with a system or component, or before starting a task.
- Describes the benefits of using one approach rather than another, or provides information about when one particular choice or tool is more appropriate than another.
- Describes how one feature, tool, or product is related to others, and how they work together or do not work together.
- Describes any restrictions that limit the circumstances in which a tool can be used successfully.
- Expands the significance of an important term beyond the scope of a glossary definition.
- Explains how and why some behavior changes as time passes or work progresses.
- Helps users form a mental picture that builds on the experience and knowledge that they are already likely to have.

Remember these main points when you write concept topics:

- Document the background knowledge that users need to successfully use the system, process, product, tool, or feature.
- Use a noun or noun phrase as the heading.
- Use paragraphs.
- If the topic is longer than two screens of information, use subheadings to break it into sections.
- When introducing a new term, begin with a definition; then, expand that definition.

- Add graphics when they simplify the explanation, for example, to show a process or the relationship among concepts.
- Provide examples to bridge from unknown knowledge to known.
- Address only one complete idea.
- Keep the concept topic short and concise, but describe the concept completely.

Reference topics

Reference topics provide quick access to information that users are likely to need as they complete tasks. Use reference topics to document the purpose of an element, and any restrictions (such as case sensitivity), required authorizations, or anything else that might limit the use of an element.

There are several types of elements for which you can provide reference information. A few examples follow:

- APIs
- Commands
- Language and programming elements
- Class descriptions
- Keyboard shortcuts
- Protocols
- Schemas
- Settings
- Symbols
- Templates
- Column, field, and action descriptions

Remember these main points when you write reference topics:

- Use tables and lists to make reference information easy to scan.
- Use a noun or noun phrase as the title for a reference topic.
- For a particular category of information, such as API documentation, use a consistent format so users can find information quickly.
- Be brief, but write in full sentences.
- Do not go into long explanations; assume that readers understand the basic technology.
- Make the topic as long as it takes to explain the subject.
- Provide links to closely related reference topics, and in some cases, to related concept and task topics.

Developing message help

A message is any communication that is passed from the application to a user or to another application. z/OSMF uses messages to inform users of important events, such as state changes and errors that require resolution.

For the messages your application issues, use the XHTML tagging language to create a separate help file for each message. Include the following attributes for each message:

- **message ID.** Specifies a unique alphanumeric identifier that provides a quick means to distinguish one message from others. Use the message ID as the filename and topic title.

Vendor message IDs can start with the letters J-Z; the letters A-I are reserved for IBM. The last character indicates the severity of the message, which can have one of the following values:

- **I for information.** Describes information or status for normal conditions and operations.

- **W for warning.** Alerts users to a condition that might cause problems in the future. When a warning message is displayed, users can generally continue with their tasks, but those tasks might not complete in a way that is expected.
- **E for error.** Alerts users to a problem that already occurred. Users or systems cannot continue their tasks.

For example, `YYZR134I`.

- **message text.** Briefly describes the problem or situation from the user's perspective. Messages can have variables (or arguments), which are typically numbers or placeholders that are used in a message in place of a specific file name, command, component, or other object. In the message text, ensure that you:
 - Focus on the problem, not the error.
 - Describe the problem briefly, use full sentences, and ensure that the information is accurate.
 - Avoid wording that seems to blame the user.
 - Do not concatenate multiple messages to create a single message.
 - Make variables meaningful and unique.
 - Replace variables only with proper nouns.
 - Use double quotation marks around variables only when necessary.
- **explanation.** Expands the message text and provides more detail. In the explanation, ensure that you:
 - Explain why the message was issued.
 - For error and warning messages, describe the cause of the problem (when and where the error occurred), explain the consequences of the error, and provide information to help users avoid the problem in the future if possible.
 - Do not repeat the message text in the explanation section.
 - Avoid using codes to build messages and resolve problems in error handling. However, if your application uses codes, describe the codes consistently in your messages and include corrective action so users do not need to look elsewhere for an explanation of the code.
- **userResponse** and **sysprogResponse.** Describes what the user (userResponse) or system programmer and administrator (sysprogResponse) must do to proceed, to recover from the error, or to prevent a problem. If no response is required, enter *No action is required*. For warning and error messages, a response must be provided for the system programmer, the user, or both.

In the response, ensure that you:

- Use active voice when possible.
- Provide complete and specific instructions to resolve the problem.
- Link to other information if necessary.
- Categorize the actions.
- Do not leave the response section empty.
- Ensure that wrapped, associated, and stacked messages are consistently presented.

Checklist for writing effective messages

User perceptions of a software product are strongly influenced by how well messages convey relevant information and help the user solve a problem. Use the following checklist to ensure that messages are clear, accurate, complete, and helpful.

- Write accurate messages:
 - ___ Ensure that all facts in the message are accurate.
 - ___ Avoid product names and versions because they change over time.
- Avoid telling users to call a support organization or system administrator:
 - ___ Check whether there are alternative solutions.

- __ Ask users to check documentation, Web sites, and so on to find solutions.
- Do not blame the user:
 - __ Write messages so that they do not appear to blame the user, even if it the cause is a user error.
 - __ Avoid "doomsday" phrases such as *catastrophic failure* or *fatal error*.
- Use clear language:
 - __ Check for ungrammatical or incorrectly punctuated sentences.
 - __ Avoid garbled, long (over 25 words), or convoluted sentences.
 - __ Check for unnecessary passive voice.
 - __ Avoid abstract language, general language, and jargon.
 - __ Use a consistent style and word usage.
 - __ Always use full sentences with correct punctuation.
 - __ Use nouns after command and API names, for example, the `BaseException` class. (class is the noun.)
 - __ Do not leave out articles such as *a*, *an*, or *the*.
- Provide complete information:
 - __ Ensure that each message is needed in the system. Do not display a message when the code should handle the issue.
 - __ Provide all the instructions for resolving a problem in one message whenever possible.
 - __ If the user actions are too long or topics for the appropriate corrective actions exist elsewhere, provide a link from the message to other appropriate topics.
 - __ Add explanation, user response, and system programmer response sections to all messages.
 - __ Do not give simplistic user actions such as "See the log files" without more guidance. For example, if you ask users to see log files, tell them where to look in the directory structure to find the log and what to look for in the log.
 - __ Provide examples for commands, APIs, or other code unless the code is extensive or complicated.
 - __ Do not expose unnecessary information such as documenting system actions that do not affect the user, providing information that is too detailed for the target audience, and describing internal workings that the user has no control over. This information should be logged only in a trace for support.

Creating the table of contents

A table of contents is required to integrate each help plug-in into the z/OSMF help system. Providing a table of contents improves the navigability of help plug-ins and ultimately makes it easier for users to find relevant help information. This topic describes the structure of the table of contents for panel and message help plug-ins.

Panel help plug-ins

To create a table of contents for panel help plug-ins, create an XML file that has the following structure:

```
<toc label="task-name" link_to="category" topic="path-to-parent-topic">
  <topic label="task-name" href="path-to-parent-topic">
    <topic label="topic-name" href="path-to-topic"/>
    <topic label="topic-name" href="path-to-topic">
      <topic label="topic-name" href="path-to-topic"/>
      <topic label="topic-name" href="path-to-topic"/>
      <topic label="topic-name" href="path-to-topic"/>
    </topic>
    <topic label="topic-name" href="path-to-topic"/>
  </topic>
</toc>
```

Figure 544. Table of contents template for panel help plug-ins

where:

task-name

Name that will be displayed in the z/OSMF navigation area for your application followed by the word *task*.

category

Path to the category that will contain the help for the task. For more details, see [“Categorizing help plug-ins”](#) on page 1265.

path-to-parent-topic

Path to and name of the file that introduces the task and its key features. The introductory topic must be the container (parent) for all the other help topics in the plug-in.

topic-name

Name of the topic, which is the label that will be displayed in the table of contents in the z/OSMF help system.

path-to-topic

Path to and name of the file that contains the help content.

Figure 545 on page 1264 provides a sample table of contents for the System Status task.

```
<toc label="System Status task"
link_to="./com.ibm.zosmfcore.performance.help.doc/
izuG00hpPerformance.xml#sysplex"
topic="izuR00hpSysplexStatusTask.html">
  <topic label="System Status task" href="izuR00hpSysplexStatusTask.html">
    <topic label="Managing system resources" href="izuR00hpSysplexStatusPanel.html">
      <topic label="Adding resource entries" href="izuR00hpAddSysEntry.html">
        <topic label="Add and Modify Entry pages" href="izuR00hpAddModSysPanel.html"/>
      </topic>
      <topic label="Modifying resource entries" href="izuR00hpModSysEntry.html"/>
      <topic label="Removing resource entries" href="izuR00hpRemSysEntry.html"/>
    </topic>
  </topic>
</toc>
```

Figure 545. Sample table of contents for the System Status task

Message help plug-ins

To create a table of contents for message help plug-ins, create an XML file that has the following structure:

```
<toc label="message-ID-range" link_to="category" topic="path-to-parent-topic">
  <topic label="message-ID-range" href="path-to-parent-topic">
    <topic label="message-ID" href="path-to-topic">
      <topic label="message-ID" href="path-to-topic"/>
      <topic label="message-ID" href="path-to-topic"/>
      <topic label="message-ID" href="path-to-topic"/>
    </topic>
  </topic>
</toc>
```

Figure 546. Table of contents template for message help plug-ins

where:

message-ID-range

Range of message IDs included in the message help plug-in.

category

Path to the z/OSMF messages category, which will contain the message help for the task. For more details, see [“Categorizing help plug-ins”](#) on page 1265.

path-to-parent-topic

Path to and name of the file that is the container (parent) for all of the message help topics.
The parent topic should state the following: *This topic describes the z/OSMF messages that have a message ID between message-ID-range.*

topic-name

Name of the topic, which is the ID of the message.

path-to-topic

Path to and name of the file that contains user assistance for the message.

Figure 547 on page 1265 provides a sample table of contents for messages issued by z/OSMF core.

```
<toc label="IZUG0400-IZUG9999"
  link_to=" ../com.ibm.zosmfmessages.help.doc/
izuAllhpzOSMFMessages.xml#core_messages"
  topic="IZUG0400-IZUG9999.html">
  <topic label="IZUG0400-IZUG9999" href="IZUG0400-IZUG9999.html">
    <topic label="IZUG400I" href="IZUG400I.html"/>
    <topic label="IZUG401E" href="IZUG401E.html"/>
    <topic label="IZUG401W" href="IZUG401W.html"/>
    <topic label="IZUG402W" href="IZUG402W.html"/>
    <topic label="IZUG403E" href="IZUG403E.html"/>
    <topic label="IZUG404I" href="IZUG404I.html"/>
  </topic>
</toc>
```

Figure 547. Sample table of contents for the messages issued by z/OSMF core

When the message help plug-in is displayed within the z/OSMF help system, it is listed under the z/OSMF messages category and each message topic is nested under the parent topic.

Categorizing help plug-ins

In the z/OSMF help system, a task and its panel help plug-in are included in the same category, and the task's message help plug-in is listed under the z/OSMF messages category, where all z/OSMF messages are listed. This structure ensures consistency across z/OSMF tasks, making it easier for users to find relevant content.

To categorize the panel and message help plug-ins, on the <toc> element in the *toc.xml* file, specify the appropriate URL for the *link_to* attribute. Table 613 on page 1265 lists the URL to use for each z/OSMF category. For example, to add the message help plug-in to the z/OSMF messages category, type the following value: *link_to=" ../com.ibm.zosmfmessages.help.doc/izuAllhpzOSMFMessages.xml#messages"*.

Table 613. URL to use for each z/OSMF category	
Category	URL
Commands and Logs	../com.ibm.zosmfcore.commandlog.help.doc/izuG00hpCommandLogs.xml#commands_and_logs
Configuration	../com.ibm.zosmfcore.configuration.help.doc/izuG00hpzOSMFConfiguration.xml#configuration
Jobs and Resources	../com.ibm.zosmfcore.jobresources.help.doc/izuG00hpJobResources.xml#jobs_and_resources
Links	../com.ibm.zosmfcore.linksuser.help.doc/izuG00hpLinksUser.xml#links
Performance	../com.ibm.zosmfcore.performance.help.doc/izuG00hpPerformance.xml#performance
Problem Determination	../com.ibm.zosmfcore.problemdetermination.help.doc/izuAllhpzOSMFProbDet.xml#problem_determination
Software	../com.ibm.zosmfcore.software.help.doc/izuG00hpSoftware.xml#software
z/OSMF Administration	../com.ibm.zosmfcore.administration.help.doc/izuG00hpzOSMFAdministration.xml#administration
z/OSMF messages	../com.ibm.zosmfmessages.help.doc/izuAllhpzOSMFMessages.xml#messages
z/OSMF Settings	../com.ibm.zosmfcore.settings.help.doc/izuG00hpSettings.xml#settings

Identifying the contents of help plug-ins

Every plug-in requires a file called *plugin.xml* to identify the plug-in contents to the z/OSMF help system.

About this task

This file includes the following items:

- The table of contents file that contributes to the navigation for the help plug-in.
- The name, ID, and the version of the help plug-in.
- The index file for building the Index view.

Procedure

To create the *plugin.xml* file for your help plug-ins, follow these steps:

1. Create a new Extensible Markup Language (XML) file called *plugin.xml*.
2. Make the following XML declaration the first two lines of the file:

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
```

3. Add the root `<plugin>` element and specify the following attributes for the help plug-in:

name

For the panel help plug-in, the name that will be displayed in the z/OSMF desktop for your application, followed by the word *task*.

For the message help plug-in, the range of message IDs included in the plug-in.

id

Name of the folder that contains the panel or message help contents.

version

Version of the help plug-in.

provider-name

Name of your company.

For example, the plug-in identification metadata for the Incident Log task can be as follows:

```
<plugin name = "Incident Log task"
        id = "com.ibm.zosmfincidentlog.help.doc"
        version = "2.1"
        provider-name = "IBM">
```

4. Within the context of the `<plugin>` element, create an `<extension>` element with a *point* attribute value of *org.eclipse.help.toc*. Then, within the context of the `<extension>` element, create a `<toc>` element to declare the table of contents file.

```
<extension point="org.eclipse.help.toc">
  <toc file="toc.xml" primary="true" />
</extension>
```

5. If you indexed your content, within the context of the `<plugin>` element, create an `<extension>` element with a *point* attribute value of *org.eclipse.help.index*. Then, within the context of the `<extension>` element, create an `<index>` element to declare the index file.

```
<extension point="org.eclipse.help.index">
  <index file="index.xml" />
</extension>
```

6. Close the `<plugin>` element.

```
</plugin>
```


Example

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>

<plugin name = "Incident Log task"
      id = "com.ibm.zosmfincidentlog.help.doc"
      version = "2.1"
      vendor-name = "IBM">

  <extension point="org.eclipse.help.toc">
    <toc file="toc.xml" primary="true" />
  </extension>

  <extension point="org.eclipse.help.index">
    <index file="index.xml" />
  </extension>

</plugin>
```

Adding links to help plug-ins

z/OSMF tasks provide a help link, help button, or message ID link on every page, window, or message that is provided in the user interface. When users click the corresponding widget, user assistance is displayed for the current context.

Procedure

To provide context-sensitive help for your applications, complete the steps that follow:

1. Create a help topic for each page, window, and message that can be displayed in your applications. For instructions, see [“Creating help plug-ins” on page 1258](#).
2. Add a help link, help button, message ID link, or another widget to each page, window, and message.
3. For each widget, specify the URL for the related help topic. The URL must have the following format:

```
https://host:port/context-root/helps/release-id/help-plugin-name/help-topic
```

where:

host

Hostname or IP address of the system where z/OSMF is installed.

port

Secure application port for the z/OSMF configuration. If you specified a secure port for SSL encrypted traffic during the configuration process (through the IZUPRMxx parmlib member keyword HTTP_SSL_PORT), specify the same port in the URL. If you omit the port, it is assumed that you are using port 443, the default.

context-root

Context root of the z/OSMF application. By default, the context root is zosmf.

release-id

Value that identifies the release, for example SSB2H8_2.4.0.

help-plugin-name

Name of the help plug-in that contains the help topic to which you want to link.

help-topic

Filename of the help topic to which you want to link.

Example

Figure 548 on [page 1268](#) provides sample code for linking a button to a specific help topic in the z/OSMF help system.

```

<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    function open_help()
    {
      window.open(
        "https://abc.com:81/zosmf/helps/SSB2H8_2.4.0/com.ibm.task.help.doc/
about.html", "help"
      );
    }
  </script>
</head>
<body>
  <input value="Help" onclick="open_help()" type="button">
</body>
</html>

```

Figure 548. Sample code for linking to a help topic

Adding your applications to z/OSMF

To add your application to z/OSMF, create a property file that defines the parameters required for z/OSMF to configure your application, and use the z/OSMF Import Manager task to import the property file.

Before you begin

- Develop a web-based application and the supporting documentation for the functions you want to add to z/OSMF. For instructions, see [“Developing web-based applications”](#) on page 1226 and [“Authoring end user assistance”](#) on page 1257.
- Identify and resolve any security vulnerabilities in your application. For more details, see [“Verifying the security of applications”](#) on page 1274.

About this task

A property file is a flat file, such as a text file, that contains a set of attributes for one or more instances of an object. The attributes are specified as name and value pairs, and must be enumerated for each instance of the object. z/OSMF supports property files that are encoded using the platform default encoding, which by default is EBCDIC 1047, or ASCII.

When you use a property file to import a plug-in into z/OSMF, the z/OSMF server performs the following actions:

- Validates the structure of the property file, and verifies that a value is provided for the required properties.
- Creates symbolic links to the client-side code for your application so that any updates to that code are reflected in z/OSMF.
- Stores the symbolic links in the plug-in context root subdirectory, which is specified in the plug-in property file.
- Adds a task that is defined in the property file to the z/OSMF desktop, if a URL is specified for its `taskNavigationURL` property.

Procedure

To use a property file to add a plug-in to z/OSMF, complete the following steps:

1. Create a property file in the z/OS UNIX System Services (z/OS UNIX) file system, and set 644 as its permissions.
For example, you can create a text file called *plugin.properties*.

2. Type `importType=plugin` at the beginning of the file. Doing so informs z/OSMF that the property file contains a plug-in definition. You cannot define other types of objects, such as event types or handlers, in a property file for a plug-in.
3. Define only one plug-in to be added to z/OSMF. To do so, specify the following properties for the new plug-in:

```
izu.externalapp.file.version=plugin-version
izu.externalapp.local.context.root=plugin-context-root
izu.externalapp.code.root=plugin-code-root

pluginId=plugin-Id
pluginDefaultName=plugin-name
pluginDescription=plugin-description
aboutPanelPath=about-panel-path

izu.externalapp.help.root=help-root
izu.externalapp.helpdoc=help-plugin-name
```

plugin-version

Specify the version of the plug-in.

plugin-context-root

Specify the name of the symbolic link directory under the `USERDIR/data/externalapps` directory to use for your applications. The `USERDIR` setting, which identifies the mount point of the z/OSMF user file system, is set when your installation configures z/OSMF. By default, the mount point is `/global/zosmf/`.

If you specify `myapp` as the plug-in context root, a symbolic link that is named `myapp` is created in the path of `USERDIR/data/externalapps/myapp`, and it points to the source code path `/usr/lpp/myapp/ui`.

z/OSMF uses the context root to build the navigation URL and the bundle URL.

The plug-in context root is required, and the directory name must comply with the z/OS UNIX naming guidelines.

plugin-code-root

Specify the path to the z/OS UNIX directory that contains the source code for the tasks that are included in the plug-in. If the plug-in property file and the source code directory reside in the same directory, you can specify a relative path, for example, `ui`. Otherwise, you must specify the absolute path, for example, `/usr/lpp/myapp/ui`. The plug-in code root is required.

plugin-ID

Specify a unique identifier for the plug-in. The ID is required, must be unique, and can contain a maximum of 64 characters.

plugin-name

Specify the name of the plug-in. The plug-in name is required, and can contain a maximum of 64 characters.

plugin-description

Provide a description of the plug-in. The description is required, and can contain a maximum of 256 characters.

about-panel-path

Specify the absolute path of the flat file that contains the plug-in version information to display on the z/OSMF **About** page. For example, `/zosmf/IzuUICommon/externalfiles/myapp/myappVersion.txt`. The path is required.

Tip: If the file resides in the same directory as the plug-in property file, you can specify the file name with no path information. For example, `myappVersion.txt`.

help-root

Specify the path to the z/OS UNIX directory that contains the help files for the tasks that are included in the plug-in. If the plug-in property file and the help root directory reside in the same directory, you can specify a relative path, for example, `helps`. Otherwise, you must specify the absolute path, for example, `/usr/lpp/myapp/helps`. The plug-in help root is optional.

help-plugin-name

Specify the name of the directory that contains the help files for the tasks in your application. For example, `com.ibm.zosmfmyapp.task1.help.doc`. The directory name is required only if you are including help files in your plug-in. The directory name must comply with the z/OS UNIX naming guidelines.

Each help plug-in must be enumerated and must be listed in numerical order. z/OSMF expects the first help plug-in to be enumerated as *izu.externalapp.help.root.1*, the second as *izu.externalapp.help.root.2*, and so on. The first time the enumeration does not match the position of the help plug-in in the file, z/OSMF stops looking for help plug-ins.

4. Include one to 32 task definitions in the plug-in property file. Each task definition must be enumerated and must be listed in numerical order because z/OSMF expects the first task definition to be enumerated as one, the second definition as two, and so on. The first time the enumeration does not match the position of the task definition in the file, z/OSMF stops reading the file. The remaining task definitions are not processed.

The following attributes are supported for tasks:

```
taskId=task-ID
taskVersion=task-version
taskCategoryId=category-ID
taskDispName=task-name
taskDispDesc=task-description
taskSAFResourceName=task-SAF-resource-name
taskMultiSysplexScope=task-multi-sysplex-scope
taskHandlerEligible=task-handler-eligible
taskAuthenticatedGuestEligible=task-authenticated-guest-eligible
taskNavigationURL=task-navigation-URL
taskBundleUrl=bundle-URL
taskBundleFileName=bundle-file-name
taskMinZOS=minimum-z/OS-level
taskMinZOSMF=minimum-z/OSMF-level
```

task-ID

Specify a unique identifier for the task. The ID is required, must be unique, and can contain a maximum of 64 characters.

task-version

Specify the version of the task.

category-ID

This value is ignored.

task-name

Specify the task name to be displayed in the z/OSMF desktop. The task name is required, and can contain a maximum of 30 characters.

task-description

Provide a description of the task. The description is required, and can contain a maximum of 200 characters.

task-SAF-resource-name

Specify a unique SAF resource name to be used for managing user authorizations to the task. The resource name must start with ZOSMF, and must conform to the following structure to ensure uniqueness:

```
ZOSMF.<vendor>_<plugin-ID>.<task-ID>.<task-name>
```

Where:

vendor

Name of your company.

plugin-ID

Unique identifier that you assigned to the plug-in.

task-ID

Unique identifier that you assigned to the task.

task-name

Name that you assigned to the task.

For example, `ZOSMF . IBM_ TESTPLUGIN . COMMANDS . Commands`.

If the SAF resource name does not begin with ZOSMF, z/OSMF adds prefix ZOSMF.IMPORT to the SAF resource name.

The resource name is required, and can contain up to 231 alphanumeric characters (A-Z a-z 0-9) and the following special characters: Underscore (`_`), dash (`-`), period (`.`). The use of a period in a resource name is treated as a qualifier. As such, the first character after a period must be A-Z or a-z.

task-multi-sysplex-scope

z/OSMF provides a multi-sysplex capability, which allows you to manage multiple z/OS sysplexes from a single z/OSMF instance. To do so, a z/OSMF instance must be running in each sysplex to be managed.

Set the *task-multi-sysplex-scope* property to `true` to indicate that this task can be used to manage or display data for multiple z/OS sysplexes. Otherwise, omit this property or set it to `false`.

task-handler-eligible

z/OSMF provides the application linking capability, which allows you to create context-sensitive launch points between tasks or applications. The task or application that initiates the launch request is referred to as the event requester. The task or application that processes the request and displays the appropriate context is referred to as the event handler.

Set the *task-handler-eligible* property to `true` to indicate that the task is eligible to be an event handler in the application linking process. Doing so allows users to define the task as a handler for one or more event types. To disallow the task to participate in the application linking process as an event handler, omit this property or set it to `false`.

task-authenticated-guest-eligible

Set this property to `true` to extend task authorization to users who are logged in to z/OSMF, but are not defined to a z/OSMF SAF security group. Otherwise, omit this property or set it to `false`. Extending task authorization to users who are not logged in to z/OSMF is not supported.

task-navigation-URL

Specify the relative or absolute path of the home page for the task. For example, if the home page is named `index.html` and resides in the `myapp` directory (context root), you can specify `index.html` or `/zosmf/IzuUICommon/externalfiles/myapp/index.html`.

If the plug-in contains more than one task, append the following to the path: `? task=contextRoot.taskName`, where *contextRoot* and *taskName* are the values you specified for the `izu.externalapp.local.context.root` property and the `taskDispName` property. For example, `/zosmf/IzuUICommon/externalfiles/myapp/index.html?myapp.settings.task`.

The path can contain a maximum of 4000 characters, including alphanumeric characters (A-Z a-z 0-9), blanks, mathematical symbols (`+ - = | ~ () { } \`), punctuation marks (`? , . ! ; : ' " / []`), and the following special characters: `%`, `$`, `#`, `@`, `^`, `*`, and `_`. Any leading or trailing white space is ignored.

The navigation URL is required if the bundle URL is specified. Otherwise, omit the navigation URL.

bundle-URL

Specify the relative or absolute path of the language resource bundle for the task. For example, if the bundle file resides in the `/myapp/js` directory, you can specify `/js` or `/zosmf/IzuUICommon/externalfiles/myapp/js`.

The path can contain a maximum of 256 characters, including alphanumeric characters (A-Z a-z 0-9), blanks, mathematical symbols (`+ - = | ~ () { } \`), punctuation marks (`? , . ! ; : ' " / []`), and the following special characters: `%`, `$`, `#`, `@`, `^`, `*`, and `_`. Any leading or trailing white space is ignored.

The bundle URL is required if the navigation URL is specified. Otherwise, omit the bundle URL.

bundle-file-name

Specify the name of the language resource bundle file. The file name can contain a maximum of 256 characters. For example, `bundle.js`. The file name is required if the bundle URL is specified.

minimum-z/OS-level

Specify the minimum z/OS operating system level that the task requires. You can specify one of the following values:

- **04.28.00:** Indicates that the minimum z/OS level is V2R5.
- **04.27.00:** Indicates that the minimum z/OS level is V2R4.
- **04.26.00:** Indicates that the minimum z/OS level is V2R3.

minimum-z/OSMF-level

Specify the minimum z/OSMF level that the task requires. You can specify one of the following values:

- **04.28.00:** Indicates that the minimum z/OSMF level is V2R5.
- **04.27.00:** Indicates that the minimum z/OSMF level is V2R4.
- **04.26.00:** Indicates that the minimum z/OSMF level is V2R3.

[Figure 549 on page 1273](#) provides a sample property file that defines the *myapp* plug-in, which contains a Commands task and a Settings task.

```

importType=plugin

izu.externalapp.file.version=1.0.0
izu.externalapp.local.context.root=myapp
izu.externalapp.code.root=ui
pluginId=com.ibm.zosmf.myapp
pluginDefaultName=myapp
pluginDescription=Operate myapp.
aboutPanelPath=/zosmf/IzuUICommon/externalfiles/myapp/myappVersion.txt

izu.externalapp.help.root=helps
izu.externalapp.helpdoc.1=com.ibm.zosmfmyapp.commands.help.doc
izu.externalapp.helpdoc.2=com.ibm.zosmfmyapp.settings.help.doc

taskId1=COMMANDS
taskVersion1=1.0
taskCategoryId1=12
taskDispName1=Commands
taskDispDesc1=The Commands task lets you enter z/OS commands.
taskSAFResourceName1=ZOSMF.IBM_COMMANDS.COMMANDS.Commands
taskMultiSysplexScope1=true
taskHandlerEligible1=true
taskAuthenticatedGuestEligible1=true
taskNavigationURL1=/zosmf/IzuUICommon/externalfiles/myapp/index.html?
task=myapp.commands
taskBundleUrl1=/zosmf/IzuUICommon/externalfiles/myapp/js/nls
taskBundleFileName1=bundle.js
taskMinZOS1=04.27.00
taskMinZOSMF1=04.27.00

taskId2=SETTINGS
taskVersion2=1.0
taskCategoryId2=10
taskDispName2=Settings
taskDispDesc2=The Settings task allows you to define task-specific settings.
taskSAFResourceName2=ZOSMF.IBM_SETTINGS.SETTINGS.Settings
taskHandlerEligible2=true
taskAuthenticatedGuestEligible2=true
taskNavigationURL2=/zosmf/IzuUICommon/externalfiles/myapp/index.html?
task=myapp.settings
taskBundleUrl2=/zosmf/IzuUICommon/externalfiles/myapp/js/nls
taskBundleFileName2=bundle.js
taskMinZOS2=04.27.00
taskMinZOSMF2=04.27.00

```

Figure 549. Sample plug-in property file

Tip: To remove a plug-in and all of its tasks from z/OSMF, type *deletePlugin=true* at the end of the property file. Otherwise, omit this property.

5. Save the property file.
6. Optionally, you can add a customized icon for each new task.

By default, z/OSMF uses a "puzzle piece" icon to represent a task in the z/OSMF desktop. However, you can supply a customized icon for the task by following these steps:

- a. Create the icon as a Portable Network Graphics (PNG) file. For consistency with the other z/OSMF desktop icons, set the file dimensions to 470x470 pixels.
- b. Save the icon image file by using the following naming convention: *task-name.png*
- c. Store the icon image file in the same directory path as the main page for the task URL. For example, if the task name is myapp and the task main page is `/zosmf/IzuUICommon/externalfiles/myapp/index.html`, store the file as follows:

```
/zosmf/IzuUICommon/externalfiles/myapp/myapp.png
```

7. Import the property file. To do so, complete the following steps:

- a) In the z/OSMF desktop, select **Import Manager**.
- b) On the **Import** tab on the **Import Manager** page, specify the full path and name of the property file you created, and click **Import**.

Results

A message is displayed to indicate whether the plug-in was added. If so, the tasks where you provided a URL for the taskNavigationURL property are displayed in the z/OSMF desktop.

If z/OSMF finds any errors in the property file, the plug-in and its tasks are not added to z/OSMF. If so, you must resolve the errors and import the property file again.

What to do next

If the server-side code for your plug-in does not reside on the z/OSMF server, you must associate an application server with each task included in the plug-in. Otherwise, the plug-in will not work correctly. For instructions, see the topic about associating application servers with imported tasks in the z/OSMF help system.

Set up security for your plug-in. After which, you must refresh the security management product on your system and restart the z/OSMF server to have your changes take affect. For more details, see the section about securing your applications in [“Securing your applications” on page 1274](#).

Securing your applications

To secure your applications, identify and resolve any security vulnerabilities, and work with your security administrator to grant users access to your applications. When the required security controls are established on your system, a user can begin using z/OSMF to perform system management tasks.

Verifying the security of applications

Before importing applications into z/OSMF, ensure that the vendor or developer who supplied the application adhered to security best practices for Web applications. If the software installed is not secure, it is possible to expose your system or company to security issues.

Controlling access to applications

After importing your plug-in into z/OSMF, work with your security administrator to authorize users to your applications. z/OSMF security is based on the following concepts:

user authentication

When a user attempts to log in to z/OSMF through a web browser, the user’s credentials are verified by the z/OS host system through the SAF interface or a security management product (for example, RACF). This processing ensures that the user ID is known to the z/OS system, and the password is valid.

user authorization

Access to your application is controlled through SAF resource profile `<safPrefix>.<taskSAFResourceName>`, where `<safPrefix>` is configured in z/OSMF and is by default `IZUDFLT` and `<taskSAFResourceName>` is the SAF resource name you specified for the task in the plug-in property file. The SAF resource profile is defined in the `ZMFAPLA` class.

If your installation is using RACF and you want to assign administrators `CONTROL` access and users `READ` access to your application, you can create a profile like the following:

```
RDEFINE ZMFAPLA +
  (IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands) UACC(NONE)
PERMIT +
  IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands +
  CLASS(ZMFAPLA) ID(IZUADMIN) ACCESS(CONTROL)
PERMIT +
```



```
IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands +  
CLASS(ZMFAPLA) ID(IZUUSER) ACCESS(READ)
```

z/OSMF automatically manages the authorization of non-authenticated guests (not logged in) and authenticated guests (logged in, but are not defined to a z/OSMF SAF security group). By default, a non-authenticated guest user can access the z/OSMF Welcome task and access the default links. An authenticated guest can access everything a non-authenticated guest can, and also view the online help.

To authorize authenticated guest users to your task, in the plug-in property file, set the *task-authenticated-guest-eligible* property to *true*. Extending task authorization to users who are not logged into z/OSMF is not supported.

Actions for security update

Changes to your security setup require applicable refreshes of the security product and a restart of the z/OSMF server for them to take effect.

Chapter 4. Preparing software to exploit cloud provisioning

This topic describes how software providers can prepare software to exploit IBM Cloud Provisioning and Management for z/OS.

IBM Cloud Provisioning and Management for z/OS allows software consumers, from a selection of software services, to quickly provision and deprovision software as needed. For more information, see [“Cloud provisioning services” on page 46](#) and the online help for the Cloud Provisioning tasks in z/OSMF.

Software is provisioned from a software services template. A software services template requires the following files.

Workflow definition file

A workflow definition file is the primary XML file that defines the workflow that performs the provisioning. The workflow definition file includes information about the workflow, such as name and version, as well as step and variable definitions.

The provisioning workflow definition file must be located in a z/OS UNIX file. File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set.

For information about workflow definition files, see [Chapter 2, “Creating workflow definitions for z/OS,” on page 1099](#).

Variable input file

A workflow variable input file is a properties file that is used to specify values for one or more of the input variables that are defined in the workflow definition. For information, see [Chapter 2, “Creating workflow definitions for z/OS,” on page 1099](#).

Action definition file

An action definition file describes the actions that can be performed against the provisioned software, which is known as a software services instance.

The actions definition file can be located in a z/OS UNIX file or a z/OS data set. File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set.

For more information, see [“Actions definition file” on page 1279](#).

Documentation files

A documentation file is an optional text or PDF file that provides information that is important to provisioning the software. For example, it might describe the workflow and other files, and describe requirements for using them to provision software. There can be one documentation file for administrators, who create the software services template and prepare the software for provisioning, and one for consumers, who use the published software services template to provision the software. The document for administrators might indicate, for example, whether a network resource pool or WLM resource pool is required.

Manifest, or template source file

A manifest file is optional. It provides a shortcut when a user creates the software services template by using the z/OSMF Software Services task. Rather than specifying each of the files (workflow definition,

input variable file, action, and documentation) individually, the user can specify just the manifest file, then click **Load** to supply values for the other files.

The file must be in Java property file format:

- Each entry is a single line, in *property=value* or *property:value* format.
- The \ character is a continuation character so that a value can span lines.
- For newline, carriage return, and tab, use \n, \r, and \t.
- Comment characters are ! and #. Lines that start with those characters are ignored.

The fields in the manifest file are:

workflow-definition-file

Name of the workflow definition file

workflow-variable-input-file

Name of the workflow variable input file

action-definition-file

Name of the action definition file

description

Brief description of the workflow. This is optional.

admin-documentation-file

Name of the file that describes the workflow and other files. This file is intended for an administrator who prepares a software services template that consumers can use to provision the software. This is optional.

admin-documentation-type

File type of the administrator documentation file: text or pdf. This is optional, and valid only if admin-documentation-file is specified.

cconsumer-documentation-file

Name of the file that describes the workflow and other files, intended for a consumer who uses the software services template to provision the software. This is optional.

cconsumer-documentation-type

File type of the consumer documentation file: text or pdf. This is optional, and valid only if consumer-documentation-file is specified.

You can specify relative or absolute paths for the files that you identify in the manifest file, as follows.

- If the manifest file is a z/OS UNIX file, specify:
 - z/OS UNIX files with a full path, for example, /a/b/c/d.xml, or a relative path, for example, ../b/c/d.xml.
 - Data sets with // followed by a fully qualified name. Data sets can be either partitioned or sequential. For example, you might specify //IBMUSER.DS.PDS(XML) or //IBMUSER.DS.SEQ.
- If the manifest file is in a data set, specify:
 - z/OS UNIX files with a full path, for example, /a/b/c/d.xml.
 - Data sets with fully qualified or relative names, as follows.
 - Fully qualified names follow // and can be either partitioned or sequential. For example, you might specify //IBMUSER.DS.PDS(XML) or //IBMUSER.DS.SEQ.
 - Relative names vary with the type of the manifest file data set, as follows:
 - Partitioned: Specify just the member, which identifies a member in the manifest file data set. For example, if the manifest file is IBMUSER.DS.PDS(MF), specifying a file path of XML in the manifest file requests IBMUSER.DS.PDS(XML).
 - Sequential: Specify one or more qualifiers that are added to the manifest file data set name. For example, if the manifest file is IBMUSER.DS.SEQ, specifying a file path of XML in the manifest file requests IBMUSER.DS.SEQ.XML.

The following is an example of a manifest file.

```
# provision.mf
#
# Manifest file to be used when adding a template for provisioning an MQ for z/OS Queue Manager.
#
# <copyright
#   notice="lm-source"
#   pids="@PID###@"
#   years="2015,2016"
#   crc="3073404564">
#   Licensed Materials - Property of IBM
#
#   @PID###@
#
#   (C) Copyright IBM Corp. 2015, 2016 All Rights Reserved.
# </copyright>
#
# Provision Queue Manager workflow file (steps to provision a Queue Manager)
workflow-definition-file:provision.xml
# Provision Queue Manager workflow variables properties file (properties to be used when provisioning
a Queue Manager)
workflow-variable-input-file:workflow_variables.properties
# Queue Manager actions file (defines the actions that can be performed against a Queue Manager)
action-definition-file:qmgrActions.xml
# Provision Queue Manager workflow description
description:This workflow provisions an MQ for z/OS Queue Manager
# Provision Queue Manager readme file
admin-documentation-file:mqaas_readme.pdf
# Provision Queue Manager readme file type
admin-documentation-type:pdf
```

Actions definition file

An actions definition file has XML syntax and conforms to the rules of the actions schema. The schema defines the required and optional properties (XML elements and attributes). It imposes constraints on the order in which the elements are specified, and on the values that can be specified for each element and attribute.

The schema file is UTF-8 encoded.

If you are developing an actions definition file, you require access to the schema, and therefore access to the z/OS system on which z/OSMF is installed.

The primary XML file must start with a processing instruction (in column 1 of line 1) for the XML processor. This instruction defines the version of XML used and the encoding of the file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The remaining elements are as follows.

<actionList>

Is the root element. It contains the actions definitions.

<action>

Contains an action definition. There must be 1 - 50 actions in an actions definition file. The action contains either a command, workflow, or instruction element. The attributes are:

- name, which specifies the name of the action
- deprovision, which accepts true or false, to indicate whether the action is for deprovisioning. There must be at least one deprovision action.

<command>

Contains a command definition. It contains the following elements.

<commandValue>

Command to be issued. This is required.

<runAsUser>

User ID under which the action is to be performed. This is optional. The attribute is substitution, which accepts true or false.

<approver>

A user ID, or a list of user IDs separated by blanks. At least one user ID must approve the action before it is performed on behalf of the user ID that is specified with the runAsUser element. To specify multiple required approvers, use multiple approver elements (up to 12). The approver element is optional. If it is specified, then the runAsUser element is required. The action definition supports the same forms as the workflow definition. For more information and examples, see [“Specifying approvers for a step” on page 1152](#).

<unsolkey>

Key to search for in the unsolicited messages, for a command-type action.

<solkey>

Key to search for in the solicited messages command response, for a command-type action.

<detectTime>

Time in seconds to search for the unsolkey in the unsolicited messages. Also, the minimum time before a command response is checked for after the command is submitted for execution.

<workflow>

Contains a workflow definition, consisting of the elements that follow.

<cleanAfterComplete>

Indicates whether the workflow is removed if it completes successfully. Accepts true, false, and inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance. The default is inherit.

<wfDefFile>

Workflow definition file path. This element is required. The maximum length of the file path is 1024 characters.

<wfVarInFile>

Workflow variable input file path. This element is optional. The maximum length of the file path is 1024 characters.

<wfVar>

Assigns a value to a workflow instance variable defined in the action workflow. During processing of an action workflow, values for variables are obtained from the property file that is specified with element wfVarInFile. Use wfVar to assign a different value for an action workflow variable. You can:

- Specify an explicit value for the variable.
- Request that the variable value is obtained from the registry for the software services instance. Because all the variables and their values are captured in the software instance registry when the template is provisioned, using the wfVar element lets you share the selected variables between provisioning and other action workflows such as the deprovisioning action workflow.

This wfVar element is optional. Up to 1500 variables are allowed. It includes these attributes:

name

Is the name of the variable in the action workflow definition. The name must be unique.

updateRegistry

Indicates whether to update the variables in the software services instance registry from the action workflow. This is allowed only if the value for the action workflow variable is obtained from the registry. The value for updateRegistry must be true or false. The default is false. When the value is true, after the action is completed:

- If the variable already exists in the software services instance, the value for the variable is updated from the action workflow.
- If the variable does not already exist in the instance, the variable is created in the instance with the value from the workflow.

Examples: In the following example, the value for action workflow variable DFS_PORTID is obtained from the registry variable that is identified by \${DFS_PORTID}. The registry is updated with the new value for the DFS_PORTID variable set during action processing.

```
<wfVar name="DFS_PORTID" updateRegistry="true">${DFS_PORTID}</wfVar>
```

In the following example, the value for action workflow variable DFS_REGION_TCPIPPORT is explicitly set to 8080. You cannot specify updateRegistry="true" when the variable value is explicitly specified.

```
<wfVar name="DFS_REGION_TCPIPPORT" updateRegistry="false">"8080"</wfVar>
```

<instructions>

Defines an instruction.

<description>

Contains a brief description of the action and the function it performs. This element is optional.

Examples

The following illustrates the elements of an action definition.

```
<?xml version="1.0" encoding="utf-8"?>
<actionList>
  <action name="workflow1">
    <workflow>
      <wfDefFile>workflow1.xml</wfDefFile>
      <wfVar name="var1" updateRegistry="false">var1val</wfVar>
      <wfVar name="var2" updateRegistry="true">var2val</wfVar>
      <wfVar name="var3">var3val</wfVar>
    </workflow>
    <description>Description of Workflow 1</description>
  </action>
  <action name="workflow2">
    <workflow>
      <wfDefFile>workflow2.xml</wfDefFile>
    </workflow>
    <description>Description of Workflow 2</description>
  </action>
  <action name="instructions1">
    <instructions>The instructions</instructions>
    <description>For best results, read every word!</description>
  </action>
  <action name="command1">
    <command>
      <commandValue>d iplinfo</commandValue>
    </command>
    <description>What this command does</description>
  </action>
  <action name="deprovision" deprovision="true">
    <workflow>
      <wfDefFile>deprovision.xml</wfDefFile>
    </workflow>
    <description>This workflow can be deprovisioned</description>
  </action>
</actionList>
```

The following example shows definitions for deprovision, start, and stop actions.

```
<?xml version="1.0" encoding="utf-8"?>
<actionList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./actions_v1.xsd">
  <action name="deprovision">
    <workflow>
      <wfDefFile>./deprovision.xml</wfDefFile>
      <wfVarInFile>./workflow_variables.properties</wfVarInFile>
      <wfVar name="DFS_PORTID" updateRegistry="false">${DFS_PORTID}</wfVar>
    </workflow>
  </action>
</actionList>
```

```

        <wfVar name="DFS_SSLPORTID" updateRegistry="false">${DFS_SSLPORTID}</wfVar>
    </workflow>
    <description>This workflow can be deprovisioned</description>
</action>
<action name="start">
    <workflow>
        <wfDefFile>./Start.xml</wfDefFile>
        <wfVarInFile>./workflow_variables.properties</wfVarInFile>
        <wfVar name="url" updateRegistry="true">${ipadd}:${port}</wfVar>
    </workflow>
</action>
<action name="stop">
    <workflow>
        <wfDefFile>./Stop.xml</wfDefFile>
        <wfVarInFile>./workflow_variables.properties</wfVarInFile>
        <wfVar name="processID" updateRegistry="false">${pid}</wfVar>
    </workflow>
</action>
</actionList>

```

Appendix A. Enabling tracing for the z/OS jobs REST interface

For diagnostic purposes, your installation might be asked by IBM Support to enable tracing for the z/OS jobs REST interface. This topic provides instructions for enabling several commonly-used traces.

Your installation can trace the use of the z/OS jobs REST interface on the z/OSMF system. Also, you can trace a variety of JES related activities, which can result from the use of the z/OS jobs REST interface services.

Understand that tracing carries a performance cost. Do not activate tracing for z/OSMF unless directed to do so by IBM Support.

Tracing the z/OS jobs REST interface services

To trace the use of z/OS jobs REST interface services on the z/OSMF system, use the **MODIFY** command with the LOGGING option. Your user ID must be permitted to enter this operator command.

The command has the following format:

```
f server-name,logging='com.ibm.zosmf.restjobs.*=all'
```

where:

server-name

Is the server for your z/OSMF configuration. Set this value to the job name of the z/OSMF server, which is IZUSVR1, by default.

com.ibm.zosmf.restjobs.*=all

Is the trace specification for the z/OS jobs REST interface.

Enter the command from the operator console. The command output is displayed in the operator console and in the z/OS system log.

Your changes take effect immediately and remain in effect while the server is running. Your changes are discarded when the server is restarted, and the previous settings are used.

To end this level of tracing and revert to the previous setting, enter the command again, and specify "reset" as the trace specification, for example:

```
f izusvr1,logging='reset'
```

Tracing the JES related activities for your programs

For callers of the z/OS jobs REST interface, your installation can trace the JES related activities that can occur on behalf of program requests.

Specifically, you can trace the following JES related activities:

- Usage of the following subsystem interface (SSI) function codes:
 - Extended status function call (SSI function code 80), which allows a user-supplied program to obtain detailed status information about jobs and SYSOUT in the JES queue
 - Modify job function call (SSI function code 85), which allows a user-supplied program to modify job properties and to manage memory associated with the request.
- VSAM related activities.
- JES symbolic parameter substitutions.
- HSM recall activities.

To capture this type of information, you must add the appropriate trace specifications to the z/OSMF bootstrap.template file, which ensures that the server is started with the proper traces enabled. z/OSMF writes the trace output to files in the z/OSMF logs directory.

To start this type of tracing, do the following:

1. Locate the z/OSMF bootstrap.template file. By default, the location is: /etc/zosmf/servers/zosmfServers/bootstrap.template
2. Save a copy of the existing bootstrap.template file as a back-up.
3. Edit the bootstrap.template file, as needed:
 - Add the property `zosjes.logging=t` to capture information about the following activities:
 - Usage of the extended status function call (SSI function code 80)
 - VSAM related activities.
 - Add the property `izurestjobs.logging=t` to capture information about the following activities:
 - Usage of the modify job function call (SSI function code 85)
 - JES symbolic parameter substitutions
 - HSM recall activities.

A portion of the file is shown in Figure 550 on page 1284.

```
# Licensed Materials - Property of IBM
#
# "Restricted Materials of IBM"
#
# Copyright IBM Corp. 2013 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with
# IBM Corp.
#
# -----
#

izu.hostname=*
izu.https.port=443

# Trace options follow...
zosjes.logging=t
izurestjobs.logging=t
```

Figure 550. Bootstrap properties for z/OSMF

4. Save the bootstrap.template file.
5. Restart the z/OSMF server and resume z/OSMF operations.

Your changes will take effect immediately and are maintained across z/OSMF server restarts.

To work with the z/OSMF log files, you require a user ID with z/OSMF administrator authority (that is, a user ID defined to the z/OSMF administrator security group).

For information about how to enable other trace options for z/OSMF, and how to work with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

Appendix B. Creating product information files for the Software Management task

A *product information file* is a flat file, such as a text file, that contains information about one or more products. This information includes, for example, the product announce date, general availability date, and end of service date. You can create your own product information files or obtain them from a provider, such as IBM, another vendor, or a third party.

z/OSMF displays data from product information files in several views in the Software Management task. For example, this information is displayed in the Products page, the Products, Features, and FMIDs page, and the End of Service report.

Syntax for product information files

To be processed by z/OSMF, product information files must be formatted as JSON data and have the following syntax:

```
{
  "Version": "date-modified",
  "Products":
  [
    {
      "prodName": "product-name",
      "prodId": "product-identifier",
      "prodVRM": "version-release-modification",
      "GAAnnounceDate": "date-announced",
      "GADate": "general-availability-date",
      "URL": "URL",
      "EOSDate": "end-of-service-date",
      "country": "country"
    }
  ]
}
```

where,

date-modified

Date the file was created or last updated. The date must have the format YYYY-MM-DD. The date is required.

product-name

Name of the product. The name is optional, and is not used by z/OSMF. To omit the product name, exclude the field, type `null` as the value, or set the value equal to an empty string.

product-identifier

Identifier of the product. The product ID is required.

version-release-modification

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level. The version, release, and modification level are required.

date-announced

Date the vendor publicly announced the details of the product. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

general-availability-date

Date that a version or release of the product is available to all users. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

URL

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions. The URL

is optional. To omit the URL, exclude the field, type `null` as the value, or set the value equal to an empty string.

end-of-service-date

Last date on which the vendor will deliver standard support services for a given version or release of the product. This date is the general end of service date. It does not account for lifecycle extensions. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

country

Country for which the end of service date is applicable. The country is optional. To omit the country, exclude the field, type `null` as the value, or set the value equal to an empty string.

The information for each product must be contained within separate braces (`{ }`) inside the brackets (`[]`), and each set of braces must be comma separated. For a sample file that contains the information for two products, see [Figure 551 on page 1287](#).

Sample product information file

```
{
  "Version": "2011-06-30",
  "Products":
  [
    {
      "prodName": "z/OS",
      "prodId": "5694-A01",
      "prodVRM": "01.10.00",
      "GAAnnounceDate": "2008-08-05",
      "GADate": "2008-09-26",
      "URL": "http://www-03.ibm.com/systems/z/os/zos/",
      "EOSDate": "2011-09-30",
      "country": "US"
    },
    {
      "prodName": "z/OS",
      "prodId": "5694-A01",
      "prodVRM": "01.13.00",
      "GAAnnounceDate": "2011-07-12",
      "GADate": null,
      "URL": "",
      "country": "US"
    }
  ]
}
```

Figure 551. Sample product information file for the Software Management task

Working with the IBM product information file

The product information file that IBM supplies for System z® software is located at the following URL: [Product information file for IBM Z software products \(public.dhe.ibm.com/services/zosmf/JSONs/IBMProductEOS.txt\)](http://public.dhe.ibm.com/services/zosmf/JSONs/IBMProductEOS.txt).

To load the contents of the file into z/OSMF, do one of the following:

- Load directly from the URL.
- Manually download the file at the URL to your local workstation.
- Manually download the file at the URL to a z/OS data set or UNIX file that the primary z/OSMF host system can access.

When transferring the file from a workstation to a z/OS data set or UNIX file, transfer the file in binary format. To avoid errors, do not convert the file to the EBCDIC character set.

After you store the file in your desired location, to retrieve its contents, complete the steps provided in the *Retrieving product information from product information files* topic in the z/OSMF online help.

Appendix C. Understanding the Portable Software Instance descriptor file

A software instance is a collection of data sets containing installed software, and other data sets that may be associated with that installed software. The software may be SMP/E managed, in which case the collection of data sets also contains, and is described by, one or more SMP/E target and distribution zone pairs, defined by a single global zone.

A portable software instance is, exactly as the name implies, a portable form of a software instance, which can be used to simplify distribution of a software instance across a network, and can be deployed by the z/OSMF Software Management task. A portable software instance is a set of portable archive files created by the SMP/E GIMZIP service routine for each of the data sets defined to the software instance, including SMPCSI data sets with all associated SMP/E managed target and distribution libraries, and a descriptor file to describe in detail the entire originating software instance.

Portable Software Instance Descriptor File

The portable software instance descriptor file contains detailed information to describe the content of the originating software instance. It contains the information required by the z/OSMF Software Management task to perform a deployment operation on the content of the portable software instance. The portable software instance descriptor file has a file name of IZUD00DF.json, and is created by z/OSMF during the operation of an Export action on a software instance.

The content of the portable software instance descriptor file is shown in the following example:

```
{ "izud.pswi.descriptor":
{
  "version": "pswi-descriptor-version"
  "created": "yyyy-mm-ddThh:mm:ssZ",
  "gimpaflocation": "relative-path-for-GIMPAF.XML-file",
  "name": "software-instance-name",
  "description": "software-instance-description",
  "datasetmergeallowed": true | false,
  "globalzone": "csi-data-set-name",
  "zones": [ {
    "name": "zone-name",
    "type": "zone-type",
    "related": "related-zone-name",
    "csi": "csi-data-set-name"
  } ]
  "datasets": [ {
    "dsname": "data-set-name",
    "volumes": [ "volume-serial" ],
    "storclas": "storage-class",
    "dstype": "data-set-type",
    "tracks": "allocated-tracks",
    "secondary": "secondary-tracks",
    "dirblocks": "directory-blocks",
    "dirblocksused": "used-directory-blocks",
    "zonedddefs": [ {
      "zone": "zone-name",
      "dddefs": [ { "dddef": "dddef-name", "path": "unix-directory", "pathcontainedinds": "unix-directory-subset" } ]
    } ],
    "mountpoint": "UNIX-path",
    "unixdirs": [ "unix-directory" ],
    "isextendedformat": true | false,
    "recfm": "record-format",
    "lrecl": "logical-record-length",
    "blksize": "block-size",
    "used": "used-tracks-percent",
    "extents": "allocated-extents",
    "dscategory": [ "data-set-category" ],
    "mergecandidates": [ "volume.data-set-name" ],
    "mergeconflicts": [ "volume.data-set-name" ],
    "archid": "gimzip-archive-id",
  } ]
}
```

```

"smpeproducts": [{
  "prodname": "product-name",
  "prodid": "product-id",
  "release": "vv.rr.mm",
  "vendor": "vendor-name",
  "url": "product-url",
  "srels": ["srel"],
  "prodsups": [{
    "prodid": "product-id",
    "release": "vv.rr.mm"
  }],
  "features": [{
    "featname": "feature-name",
    "featid": "feature-id",
    "fmids": ["fmid-name"]
  }]
}],
"fmids": ["fmid-name"],
"products": [{
  "prodname": "product-name",
  "prodid": "product-id",
  "release": "product-level",
  "vendor": "vendor-name",
  "url": "product-url",
  "features": ["feature-name"],
  "gadate": "yyyy-mm-ddThh:mm:ssZ",
  "eosdate": "yyyy-mm-ddThh:mm:ssZ" | "NotAnnounced",
  "prodinfoversion": "yyyy-mm-ddThh:mm:ssZ"
}],
"workflows": [{
  "name": "workflow-name",
  "description": "workflow-description",
  "location": {
    "smptype": "smp-type",
    "smpname": "smp-name",
    "dsname": "workflow-dsname",
    "path": "workflow-path"
  },
  "performonhostsyste": true | false
}],
"datasetproperties": [{
  "dddefname": "dddef-name",
  "zone": "zone-name",
  "dsname": "data-set-name",
  "volume": "volume-serial",
  "dstype": "DLIB",
  "properties": [{"key": "value"}]
}],
"datasetpropertylabels": [{
  "propertyname": "property-name",
  "label": "property-label"
}],
"productproperties": [{
  "prodid": "product-id",
  "release": "product-level",
  "prodname": "product-name",
  "properties": [{"key": "value"}]
}]
}
}

```

Where:

version

Indicates the version of the portable software instance descriptor. The version value may be:

1. The initial version of the descriptor file.
2. Adds support for non-SMP/E managed product information.
3. Adds support for SMP/E Product end-of-service information.
4. Adds support for workflows.
5. Adds support for datasetproperties and productproperties.
6. Adds support for pathcontainedinds and performonhostsyste
7. Adds support for datasetpropertylabels
8. Adds support for merging data sets.

created

Indicates the date and time when the portable software instance was created, in ISO 8601 format. For example, yyyy-mm-ddThh:mm:ssZ.

gimpaflocation

Indicates the relative path to the GIMPAF.XML file for this portable software instance, relative to the location of the portable software instance descriptor file.

name

Indicates the name for the originating software instance.

description

Indicates the description of the originating software instance. This is an optional property.

datasetmergeallowed

Indicates, true or false, whether the merging of the data sets is allowed when you deploy the portable software instance. The default, if not specified, is false.

globalzone

Indicates the name of the CSI data set that contains the global zone. This is an optional property, specified only if the originating software instance describes SMP/E managed software.

zones

Indicates the list of SMP/E zones from the originating software instance. This is an optional property, specified only if the originating software instance describes SMP/E managed software.

name

Indicates the zone name.

type

Indicates the type for the zone, global, target or dlib.

related

Indicates the name of the zone's related zone, if any.

csi

Indicates the name for the CSI data set that contains the zone.

datasets

Indicates the list of data sets from the originating software instance.

dsname

Indicates the originating data set name.

volumes

Indicates the list of volume serials where the originating data set resided.

storclas

Indicates the name of the storage class where the originating data set resided. This is an optional property.

dstype

Indicates the type for the data set. Can be one of the following types:

- HFS — Hierarchical file system.
- PDS — Partitioned data set.
- PDSE — Partitioned data set extended.
- SEQ — Sequential data set.
- VSAM — VSAM data set.
- ZFS — zSeries file system.

tracks

Indicates the number of 3390-device equivalent tracks (56664 bytes/track) allocated to the data set.

secondary

The number of 3390-device equivalent tracks (56664 bytes/track) in the data set's secondary space allocation.

dirblocks

For partitioned data sets, the number of directory blocks in the data set.

dirblocksused

For partitioned data sets, the number of directory blocks in the data set that are currently used.

zoneddefs

Indicates the list of SMP/E zones and DDDEF entries that reference the data set. This is an optional property, specified only if the originating software instance describes SMP/E managed software, and if the subject dataset is referenced in the SMP/E zones.

zone

Indicates the name of an SMP/E zone that contains one or more DDDEF entries for the data set.

dddefs

Indicates the list of DDDEF entries that identify the data set.

dddef

Indicates the name of the DDDEF entry.

path

Indicates the UNIX directory identified in the DDDEF entry. Null if the DDDEF entry identifies a data set.

pathcontainedinds

The fixed portion of the UNIX path that is contained in the subject data set. This value excludes the mount point and any intermediate symbolic links that are specified in a DDDEF entry.

mountpoint

Indicates the mount point for the originating UNIX file system data set. Null if the DDDEF entry identifies a data set instead of a UNIX directory. This is an optional property, specified only if the originating software instance describes SMP/E managed software, and if the subject dataset is referenced in the SMP/E zones by a DDDEF entry with a UNIX directory.

unixdirs

A list of UNIX directories that reside in the subject data set and contain one or more workflow definition files for the software instance.

isextendedformat

Indicates, true or false, if the data set is an extended format sequential data set.

recfm

Indicates the record format. The record format can be any valid combination of the following codes:

- A — ASA printer control characters.
- B — Blocked records.
- F — Fixed-length records.
- M — Machine code printer control characters.
- S — Standard (for F) or spanned (for V); used only with sequential data sets.
- T — Track-overflow feature.
- U — Undefined format records.
- V — Variable-length records.

lrecl

Indicates the logical record length.

blksize

Indicates the block size, in bytes.

used

Indicates the percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

extents

Indicates the number of extents allocated to the data set.

dscategory

List of categories for how the data set is used. Can be one or more of the following:

- DLIB — SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL — SMP/E control data set associated with the global zone.
- SMP — SMP/E control data set.
- SMPTLIB — SMPTLIB data sets associated with the global zone.
- TARGET — SMP/E managed target library, or SMP/E control data set associated with a target zone.
- WORKFLOW — Contains one or more workflow definition files for the workflows explicitly defined to the software instance.
- OTHER — None of the above.

mergecandidates

The list of identifiers for data sets in the portable software instance that are candidates for merging into the subject data set. The unique identifier for any data set in the portable software instance is a combination of the data set's name and the volume on which it resides, in this form: volume.data-set-name. Data sets are merge candidates if they have compatible physical attributes, compatible provider defined properties, and for partitioned data sets have no member names in common.

mergeconflicts

The list of identifiers for data sets in the portable software instance that cannot be merged into another data set together with the subject data set. The unique identifier for any data set in the portable software instance is a combination of the data set's name and the volume on which it resides, in this form: volume.data-set-name. Data sets are merge conflicts if they have incompatible provider defined properties with the subject data set, or for partitioned data sets have one or more members with the same name as members in the subject data set.

archid

Indicates the archive ID value, produced by the GIMZIP service routine and specified in the GIMPAF.XML file, to identify the portable archive file for the data set.

smpeproducts

Indicates the list of software products installed in the originating software instance that are managed by SMP/E. This is an optional property.

prodname

Indicates the name for the product.

prodid

Indicates the identifier for the product.

release

Indicates the version, release, and modification level for the product, in this format: vv.rr.mm.

vendor

Indicates the name for the product's vendor. This is an optional property.

url

Indicates the URL that links to additional information about the product. This is an optional property.

srels

Indicates the system or subsystem releases on which the subject product can be installed.

prodsups

Indicates the list of products that are superseded by the subject product. This is an optional property.

features

Indicates the list of features for the subject product.

featname

Indicates the name for the feature.

featid

Indicates the identifier for the feature.

fmids

Indicates the list of FMIDs for the subject feature. It is also the list of FMIDs in the originating software instance. This list includes all FMIDs associated with one or more products and features, and all FMIDs associated with no products or features.

products

The list of products from the originating software instance that are not managed by SMP/E

product-name

Name of the product, but can be up to 64 characters.

product-ID

Identifier for the product, but can be up to 64 characters.

product-level

Release level for the product, but can be up to 64 characters.

vendor-name

Name of the vendor that provides the product, but can be up to 64 characters.

product-URL

A URL that links to additional information about the product, but can be up to 256 characters.

feature-name

List of names of features for the product, but can be up to 64 characters

gdate

The date when the product became generally available. May be null, or a date value, in ISO 8601 format.

eosdate

The last date on which the vendor will deliver standard support services for the product.

- null - The end of service date is unknown for the product.
- yyyy-mm-ddThh:mm:ssZ - The known end of service date, in ISO 8601 format.
- NotAnnounced - The end of service date has not yet been announced for the product.

proinfoversion

The version for the most recent product information file that provided information for the subject product. The version value represents the date the file was created or updated. May be null, or a date value, in ISO 8601 format.

workflows

List of workflows for the software instance.

workflow-name

Name for the workflow.

workflow-description

Description for the workflow.

location

Location of the workflow definition file for the workflow.

smp-type

The SMP/E element type for a workflow definition file that is managed by SMP/E.

smp-name

The SMP/E element name for a workflow definition file that is managed by SMP/E.

workflow-dsname

The name of the data set that contains the workflow definition file.

workflow-path

The UNIX path for a workflow definition file that is a UNIX file.

performonhosts

Indicates whether the workflow steps may be performed on the host system or on another system. This property is optional and the default value is true.

true

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

false

Indicates that the workflow steps may be performed on a system other than the z/OSMF host system on which the software instance resides.

datasetproperties

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,” on page 1297](#) for more information.

dddefname

The name of the SMP/E DDDEF entry that describes an SMP/E managed data set.

zone

The zone name where the DDDEF entry resides. The zone is specified to identify a unique data set when there are more than one DDDEF entries with the same name in different zones, and each DDDEF entry identifies a different data set in the software instance.

dsname

The name of the subject data set. The data set name is specified to identify a non-SMP/E managed data set.

volume

The volume of the subject data set. The volume is specified to identify a non-SMP/E managed data set where the volume was specified to identify an uncatalogued data set.

dstype

The usage type of the subject data set. A value of DLIB indicates the data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone.

properties

A list of one or more properties for the subject data set, specified as key-value pairs.

datasetpropertylabels

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

propertyname

The name, or key, of the existing provider defined property.

label

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

productproperties

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,” on page 1297](#) for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

prodid

The identifier for the subject product. This is required to identify an SMP/E managed product.

release

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product.

prodname

The name of the subject product. This is required to identify a non-SMP/E managed product.

properties

A list of one or more properties for the subject product, specified as key-value pairs.

Appendix D. Software Management workflow variables

You can use workflows in Software Management to perform setup and configuration tasks for a software instance. When a software instance is deployed, you can use workflows that are defined for the software instance to perform those setup and configuration tasks.

Software Management provides detailed properties about a software instance as well as the data sets and products that compose software instances. These properties are exposed to workflows as workflow variables.

Note: The support for the workflow variables is added with the PTFs for APAR PH09032.

The workflow variables that are created by Software Management for a software instance are shown as follows:

```
izud-varsversion = version
izud-created = yyyy-mm-ddThh:mm:ssZ
izud-createdby = user-id
izud-system = system-nickname
izud-fmids= ["fmid-name"]
izud-globalzone= csi-data-set-name
izud-swiuuid="uuid"
izud-zones= [{
  "izud-name":"zone-name",
  "izud-type":"zone-type",
  "izud-related":"related-zone-name",
  "izud-csi":"csi-data-set-name"
}]
izud-datasets = [{
  "izud-aliases":["alias"],
  "izud-cataloged":true | false,
  "izud-catalog":"catalog.data.set.name",
  "izud-blksize":"block-size",
  "izud-dddefs": [{
    "izud-dddef":"dddef-name",
    "izud-zone":"zone-name",
    "izud-path":"unix-directory"
  }],
  "izud-dirblocks":"requested-directory-blocks",
  "izud-dirblocksused":"used-directory-blocks",
  "izud-dscategories":["data-set-category"],
  "izud-dsname":"dataSetName",
  "izud-dstype":"dataSetType",
  "izud-extents":"allocated-extents",
  "izud-extendedformat":true | false,
  "izud-lrecl":"logical-record-length",
  "izud-mountpoint":"UNIX-path",
  "izud-recfm":"record-format",
  "izud-secondary":"secondary-tracks",
  "izud-storclas":"storage-class",
  "izud-tempcatalogalias":"temporary-catalog-alias",
  "izud-tracks":"allocated-tracks",
  "izud-unixdirs":["unix-directory"],
  "izud-used":"used-tracks-percent",
  "izud-volumes":["volser"],
  "provider-dataset-property-key":"provider-dataset-property-value"
}]
izud-catalogs = [{
  "izud-catname":"catalog-name",
  "izud-cattype":"MASTER | USER",
  "izud-catdsnprefixes":["data-set-name-prefix"],
  "izud-tempcatalogalias":"temporary-catalog-alias",
  "izud-tgtsyscattype":"MASTER | USER"
}]
izud-products= [{
  "izud-eosdate":"yyyy-mm-ddThh:mm:ssZ" | "NotAnnounced" | "Unknown",
  "izud-features": [{
    "izud-featname":"feature-name",
    "izud-featid":"feature-id",
    "izud-fmids":["fmid-name"]
  }],
}
```

```

"izud-gadate": "yyyy-mm-ddThh:mm:ssZ",
"izud-prodname": "product-name",
"izud-prodid": "product-id",
"izud-prodinfoversion": "yyyy-mm-ddThh:mm:ssZ",
"izud-prodsups": [{
  "izud-prodid": "product-id",
  "izud-release": "vv.rr.mm"
}],
"izud-release": "product-level",
"izud-srels": ["srel"],
"izud-url": "product-url",
"izud-vendor": "vendor-name",
"provider-product-property-key": "provider-product-property-value"
}]

```

Where:

izud-varsversion

Indicates the version of the workflow variables that are defined by z/OSMF Software Management. izud-varsversion is an integer type variable. The version value can be:

1. The initial version of the variable set.

izud-created

Indicates the date and time when the workflow instance was created. izud-created is a string type variable.

izud-createdby

The user ID for the user that created the workflow instance. izud-createdby is a string type variable.

izud-system

The nickname of the z/OSMF host system where the software instance data sets reside. For Deployment Perform workflows, this is the target system of the deployment operation. izud-system is a string type variable.

izud-fmids

The list of FMIDs in the software instance. This includes all FMIDs associated with one or more products and features, as well as all FMIDs associated with no products or features. izud-fmids is an array type variable.

izud-globalzone

Indicates the name of the CSI data set that contains the global zone for the software instance. izud-globalzone is a string type variable.

izud-swiuuid

Indicates the universally unique identifier of the software instance. izud-swiuuid is a string type variable.

izud-zones

Indicates the list of SMP/E zones for the software instance. izud-zones is an array type variable.

izud-name

Indicates the zone name.

izud-type

Indicates the type for the zone, global, target, or dlib.

izud-related

If the zone has a related zone, then this indicates the name of the zone's related zone.

izud-csi

Indicates the CSI data set name that contains the zone.

izud-datasets

The list of data sets in the software instance. izud-datasets is an array type variable.

izud-aliases

A list of alias names for the data set.

izud-cataloged

Indicates, true or false, whether the data set is cataloged.

izud-catalog

For cataloged data sets, the name of the catalog that includes the data set.

izud-blksize

The data set block size.

izud-dsname

The data set name.

izud-dstype

The type for the data set. Can be one of the following types:

- HFS — Hierarchical file system.
- PDS — Partitioned data set.
- PDSE — Partitioned data set extended.
- SEQ — Sequential data set.
- VSAM — VSAM data set.
- ZFS — zSeries file system.

izud-dscategory

List of categories for how the data set is used. Can be one or more of the following:

- DLIB — SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL — SMP/E control data set associated with the global zone.
- SMP — SMP/E control data set.
- SMPTLIB — SMPTLIB data sets associated with the global zone.
- TARGET — SMP/E managed target library, or SMP/E control data set associated with a target zone.
- WORKFLOW — Contains one or more workflow definition files for the workflows that are explicitly defined to the software instance.
- OTHER — None of the above.

izud-volumes

The list of volume serials where the data set resides.

izud-storclas

The name of the storage class where the data set resides.

izud-tempcatalias

The name of the temporary catalog alias used to uniquely identify the data set from the z/OS driving system catalog. This value is the temporary catalog alias that is used for the alternate master catalog and is a data set name prefix to reference the data set from the driving system. To reference the data set from the driving system catalog, use a name that is constructed like this: izud-tempcatalias.izud-dsname.

If a data set is cataloged in the new alternate master catalog indirectly (using a volume symbol instead of a specific volume serial) or not cataloged at all, then the izud-tempcatalias value is null. Such data sets cannot be referenced by name from the driving system catalog. They must be referenced using the volume instead.

If not creating an alternate master catalog izud-tempcatalias is null.

izud-tracks

The number of 3390-device equivalent tracks (56664 bytes/track) allocated to the data set.

izud-dddefs

Indicates the list of SMP/E zones and DDDEF entries that reference the data set.

izud-dddef

Indicates the name of the DDDEF entry.

izud-zone

Indicates the name of an SMP/E zone that contains the DDDEF entry.

izud-path

Indicates the UNIX directory that is identified in the DDDEF entry.

izud-dirblocks

For partitioned data sets, the number of directory blocks in the data set.

izud-dirblocksused

For partitioned data sets, the number of directory blocks in the data set that is currently used.

izud-mountpoint

Indicates the mount point for the UNIX file system data set.

izud-unixdirs

A list of UNIX directories that reside in the subject data set and contain one or more workflow definition files for the software instance.

izud-extendedformat

Indicates, true or false, whether the data set is an extended format sequential data set.

izud-recfm

Data set record format.

izud-secondary

The number of 3390-device equivalent tracks (56664 bytes/track) in the data set's secondary space allocation.

izud-lrecl

Data set logical record length.

izud-used

Indicates the percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

izud-extents

The number of extents that are allocated to the data set.

provider-dataset-key and provider-dataset-property-value

A property key and its value that is defined by the provider of the software instance for the subject data set. Each property that is defined for the data set is represented as a unique key-value pair. See [“Add a new software instance” on page 476](#) for more information about the datasetproperties.

izud-catalogs

The list of catalogs where data sets in the software instance are cataloged. izud-catalogs is an array type variable.

izud-catname

Name of the catalog.

izud-cattype

Indicates the catalog type, MASTER or USER, from the z/OS driving system perspective. **Note:** When creating a new alternate master catalog, from the z/OS driving system the new alternate master catalog is a user catalog. It is a master catalog only to the z/OS target system after that system is IPLed.

izud-catdsnprefixes

A list of data set name prefixes for data sets cataloged in the subject catalog.

izud-tempcatalias

When creating an alternate master catalog, izud-tempcatalias indicates the temporary catalog alias that is used to uniquely identify the data sets in the subject catalog from the z/OS driving system catalog.

If not creating an alternate master catalog, or the saved configuration information does not match the actuals, then z/OSMF might not be able to determine the temporary catalog alias. Therefore, izud-tempcatalogalias will be null.

izud-tgtsyscattype

When creating an alternate master catalog, izud-tgtsyscattype indicates the intended catalog type for the subject catalog, MASTER or USER, from the z/OS target system perspective.

If not creating an alternate master catalog, or the saved configuration information does not match the actuals, then z/OSMF might not be able to determine the intended catalog type. Therefore, izud-tgtsyscattype will be null.

izud-products

The list of software products in the software instance. This list includes both SMP/E and non-SMP/E managed products. izud-products is an array type variable.

izud-prodname

The name for the product.

izud-prodid

The identifier for the product.

izud-release

The release level for the product.

izud-vendor

The name of the vendor that supplied the product.

izud-url

The URL that links to additional information about the product.

izud-srels

The SMP/E system or subsystem releases on which the product can be installed.

izud-prodsups

The list of products that are superseded by the subject product.

izud-prod-id

The identifier for the superseded product.

izud-release

The release level for the superseded product.

izud-features

The list of features for the product.

izud-featname

The name of the feature.

izud-featid

The identifier for the feature.

izud-fmids

The list of FMIDs for the feature.

izud-gadate

The date when the product became generally available.

izud-eosdate

The last date on which the vendor delivers standard support services for the product.

izud-prodinfoversion

The version for the most recent product information file that provided information for the subject product.

provider-product-key and provider-product-property-value

A property key and its value defined by the provider of the software instance for the subject software product. Each property defined for the product is represented as a unique key-value pair. See [“Add a new software instance” on page 476](#) for more information about the productproperties.

Provider Defined Properties

As a software provider you can define properties that are associated with individual data sets and products within a software instance. These properties are available to a workflow created by Software Management as workflow variable properties. These properties are merged with those defined by Software Management to describe the data sets and products within a software instance to create the workflow variables previously described.

Provider defined properties can be specified when you use the Software Management REST services to add or modify a software instance. Refer to the [“Add a new software instance”](#) on page 476 and [Modify software instance](#) REST services to learn how to specify provider properties. Provider defined properties are merged with the Software Management defined properties to create workflow variables. These variables are made available to a workflow when Software Management creates a new workflow instance for the software instance.

Note: The support for provider defined properties is added with the PTFs for APAR PH11650.

Example

If a data set in a software instance must be APF authorized for the software to run properly, as the provider of the software, you can create a workflow to help your users ensure that the data set is APF authorized when the software instance is deployed. Your workflow needs to know which data sets in the software instance must be APF authorized, therefore, you can define a property for the subject data sets and the workflow can reference this property in the workflow variables.

When you define the software instance by using the REST service to add a new software instance, as described in [“Add a new software instance”](#) on page 476, specify a data set property by using *datasetproperties* in the request content like this:

```
POST /zosmf/swmgmt/swi HTTP/1.1
Host: pev084.yourco.com
Content-Type: application/json
Accept-Language: en

{"swiname": "mySWI",
 "system": "PEV084",
 "globalzone": "JOHNDOE.GLOBAL.CSI",
 "targetzones": ["TGTZ"],
 "datasetproperties": [
   {"dddefname": "ABCMOD",
    "properties": [{"abc-ApfLst": "yes"}]}
 ]
}
```

Figure 552. Using the Add software instance REST API to specify a data set property in the request content

In this example the data set that must be APF authorized is managed by SMP/E and is identified by a DDDEF entry named ABCMOD. A property with a key "abc-ApfLst" and value "yes" are defined for the subject data set.

When Software Management creates a workflow instance for the software instance, workflow variables are generated, which provides the workflow programmatic access to the information about the data sets in the software instance. The workflow variable *izud-datasets* is a generated array variable containing one entry for each data set in the software instance. The entry in the *izud-datasets* array for the subject data set will contain properties like this:

```

izud-datasets = [
  ...
  { "izud-dsname": "ABC.ABCMOD",
    "izud-volumes": ["LV1234"],
    "izud-dstype": "PDS",
    ...
    "abc-ApfLst": "yes"
  },
  ...
]

```

Figure 553. Sample entry in the list for the subject data set

The data set property that you specified when you defined the software instance, abc-ApfLst, is set for the subject data set in the *izud-datasets* array entry. A template step in your workflow can analyze the entries in the *izud-datasets* array variable to determine which data sets in the software instance have this property, like this:

```

<template>
<inlineTemplate substitution="true">
APF FORMAT(DYNAMIC)
## For each data set in the software instance,
## find the data sets that must be APF authorized.
##
#foreach($dataset in ${instance-izud-datasets})
#if(${dataset.abc-ApfLst} == "yes")
APF ADD DSNAME(${dataset.izud-dsname}) VOLUME(${dataset.izud-volumes[0]})
#end
#end
</inlineTemplate>
<saveAsDataset substitution="true">
${instance-izud-createdby}.PARMLIB(PROGXX)</saveAsDataset>
</template>

```

Figure 554. Sample template step to determine the specified property

This example workflow template step creates a PROGxx parmlib member snippet for the data sets with the abc-ApfLst property set to yes. Workflow template steps such as this can be created for many other actions.

Sample workflow definition

A sample workflow definition file named IZUDWFVR is provided in the SYS1.SAMPLIB data set. It demonstrates how to create a workflow that references the workflow variables that are defined by Software Management.

Appendix E. Accessibility

Accessible publications for this product are offered through [IBM Documentation \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the [Contact the z/OS team web page \(www.ibm.com/systems/campaignmail/z/zos/contact_z\)](http://www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle, its affiliates, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names might be trademarks or service marks of others.

Index

A

- About this document [lix](#)
- accessibility
 - contact IBM [1305](#)
- actions
 - get responses for REST API [380](#)
 - instances REST API [371](#)
- adding plug-ins [1268](#)
- Application Linking Manager interface
 - description [8](#)
 - event handlers [10](#)
 - event requestors [10](#)
 - event types [10](#)
 - obtaining a list of handlers for an event type [30](#)
 - obtaining a list of the tasks that are eligible to be handlers [29](#)
 - registering an event handler [26](#)
 - registering an event type [24](#)
 - unregistering an event handler [31](#)
 - unregistering an event type [32](#)
- application programming interface for z/OSMF [1](#)
- application server routing services
 - deleting data from an application server [43](#)
 - editing data for an application server [40](#)
 - modifying data for an application server [40](#)
 - removing data from an application server [43](#)
 - retrieving data from an application server [36](#)
 - updating data for an application server [40](#)
- Application server routing services
 - overview [33](#)
- approver element [1149](#)
- approving
 - templates REST API [275](#), [276](#)
- archiving
 - templates REST API [266](#)
- assistive technologies [1305](#)
- authenticate the user with a JSON web token [1014](#)
- autoEnable attribute [1143](#)
- automated step [1143](#)
- automation-info object [1038](#), [1086](#)

C

- caller scope variable [1169](#)
- change a user password or passphrase [1016](#)
- command
 - get detection results in unsolicited messages [788](#)
 - get response [782](#)
 - issue [765](#)
- condition attribute [1137](#), [1147](#)
- conditional step

- conditional step (*continued*)
 - description [1100](#), [1117](#)
 - using [1137](#), [1147](#)
- console
 - get a command response [782](#)
 - get detection results [788](#)
 - issue a command [765](#)
- contact
 - z/OS [1305](#)
- Create z/OS UNIX zFS Filesystem
 - Create z/OS UNIX zFS Filesystem [876](#)
- creating
 - templates
 - new versions [211](#)
 - REST API [203](#)
 - creating z/OSMF plug-ins
 - Application Linking Manager Javascript APIs
 - getEventFromUrl function [1246](#)
 - getHandlers function [1244](#)
 - hasLaunchContext function [1245](#)
 - onLoadingComplete function [1250](#)
 - sendEvent function [1243](#)
 - subscribe function [1248](#)
 - communicating with z/OSMF core
 - cleanupBeforeDestroy function [1239](#)
 - cleanupBeforeDestroyComplete API [1236](#)
 - download function [1231](#)
 - editorCompare function [1230](#)
 - editorCompareWithContent API [1230](#)
 - editorWithContent function [1227](#)
 - editorWithPath function [1228](#)
 - getUserSessionId function [1236](#)
 - isContentChanged function [1237](#)
 - programmaticallyCloseTab function [1235](#)
 - shouldClose function [1238](#)
 - utilPathSelector function [1229](#)
 - developing applications [1226](#)
 - developing documentation [1257](#)
 - security [1274](#)
 - creating z/OSMF plugins
 - upload function [1232](#)

D

- data class
 - definition [597](#)
- data persistence services
 - deleting persisted user or application data [406](#)
 - deleting z/OSMF settings data [1011](#)
 - persisting user or application data [401](#)
 - retrieving persisted user or application data [403](#)
- Data persistence services
 - overview [400](#)
- data set
 - creating [837](#)
 - deleting [840](#), [842](#)

- data set (*continued*)
 - list [883, 884, 886, 889](#)
 - listing [820](#)
 - listing members [823](#)
 - reading members [826](#)
 - writing to member [831](#)

- data set member
 - list [823](#)
 - read [826](#)
 - write [831](#)

- defining plug-ins [1268](#)

- DELETE
 - service submission [989](#)

- Delete method
 - Delete z/OS UNIX zFS Filesystem [878](#)

- DELETE method
 - delete a PDS or PDSE [842](#)
 - delete a tenant [135](#)
 - delete a UNIX file or directory [865](#)
 - delete a workflow on a z/OS system [1063](#)
 - delete an archived workflow on a z/OS system [1096](#)
 - delete authentication tokens [1019](#)
 - delete data from a remote z/OSMF instance [426](#)
 - delete data from a secondary z/OSMF instance [426](#)
 - delete data from an application server [43](#)
 - delete data from one system [426](#)
 - DELETE Deployment [519](#)
 - Delete multiple CFRM policies [685](#)
 - delete persisted user or application data [406](#)
 - delete portable software instances [530](#)
 - delete response for action for a software services instance [388](#)
 - delete software instances [501](#)
 - delete software services instances [367](#)
 - delete system variables [812](#)
 - delete temporary catalog aliases [502](#)
 - delete z/OS data sets [840](#)
 - delete z/OSMF settings data [1011](#)
 - deleting data from all the system in a central processor complex (CPC) [426](#)
 - deleting data from all the system in a group [426](#)
 - deleting data from all the system in a sysplex [426](#)
 - deleting data from multiple systems [426](#)
 - end a TSO/E address space [744](#)
 - place a TSO/E address space into dormant state [744](#)
 - purging a job from the JES spool [933](#)
 - remove data from an application server [43](#)
 - remove software instances [501, 530](#)
 - unregister an event handler [31](#)
 - unregister an event type [32](#)

- delete the authentication tokens for this user [1019](#)

- Delete z/OS UNIX zFS Filesystem
 - Delete z/OS UNIX zFS Filesystem [878](#)

- deleting
 - templates
 - REST API [227](#)

- DELETE method
 - get software services templates [279](#)

E

- entity tag (ETag)

- entity tag (ETag) (*continued*)
 - using [817](#)

- error report document
 - for z/OS data set and file REST interface [893](#)

- event handler
 - obtaining a list [29, 30](#)
 - registering [26](#)
 - unregistering [31](#)

- event requestor [15](#)

- event type
 - registering [24](#)
 - unregistering [32](#)

- Example of a third party z/OSMF plug-in
 - developing applications [1226](#)

- Export
 - software instance [482](#)
- Export a defined software instance
 - Export a defined software instance [482](#)

F

- feedback
 - collecting from user of a step [1115](#)

- feedback attribute [1115](#)

- file directory
 - Create [862](#)

- file system
 - mounting [879, 881](#)

- Filesystem
 - operations [876, 878](#)

- Filesystems
 - operations [874](#)

G

- GET
 - category details [978](#)
 - details of a service from the catalog [962](#)
 - details of a service submission [990](#)
 - list of categories [975](#)
 - List the services in the catalog [956](#)
 - service submission [979](#)

- get approval
 - templates
 - REST API [269](#)

- Get method
 - Get the response to a command that was issued with the TSO/E REST API [734](#)

- GET method
 - Check MVS systems in the local plex [709](#)
 - data class definition [597](#)
 - get a command response [782](#)
 - get a domain [104, 110](#)
 - get a domain resource pool [164](#)
 - Get a list of defined administrative CFRM policies [659](#)
 - Get a notification [439](#)
 - get a published software services template [290](#)
 - get a published software services template history [302](#)
 - get a resource pool [158](#)
 - get a resource pool history [171](#)
 - get a software services instance [334](#)

GET method (*continued*)

- get a tenant for cloud provisioning [121](#)
- get a tenant history for cloud provisioning [126](#)
- Get CFRM policy properties [661](#)
- get data set attributes [76](#)
- get detection results in unsolicited messages [788](#)
- get prompt variables for a software services template [306](#)
- get response for software services instance actions [385](#)
- get responses for actions for a software services instance [380](#)
- get security profile information [194](#)
- get software services instances [345](#)
- get source information for a software services template [251](#)
- get the properties of a z/OSMF workflow [1031](#)
- get the properties of an archived workflow [1082](#)
- Get the properties of an LPAR entry [91](#)
- get the system variables [805](#)
- get variables for a software services instance [354](#), [357](#)
- get your metered data [755](#), [758](#)
- getting an approval for a software services template [269](#)
- list data classes [580](#)
- list domain resource pools [179](#)
- list LPAR resource pool entries [87](#)
- List MVS subsystems [436](#)
- list resource pools [173](#)
- list storage classes [614](#)
- list storage groups [549](#)
- list template resource pools [186](#)
- list the archived workflows for a system [1080](#)
- list the domains for cloud provisioning [112](#)
- list the jobs for an owner, prefix, or job ID [909](#)
- list the spool files for a job [912](#)
- list the tenants for cloud provisioning [129](#)
- list the workflows for a system [1054](#)
- list volumes [571](#)
- list z/OS data set members [823](#)
- list z/OS data sets [820](#)
- List z/OS UNIX Filesystems [874](#)
- listing approvals for a software services template [272](#)
- listing UNIX files and directories [850](#)
- obtain a list of groups [692](#)
- obtain a list of handlers for an event type [30](#)
- obtain a list of portable software instances [507](#)
- obtain a list of software instances [461](#)
- obtain a list of sysplexes [697](#)
- obtain a list of systems [689](#)
- obtain a list of systems in a central processor complex (CPC) [701](#)
- obtain a list of systems in a group [694](#)
- obtain a list of systems in a sysplex [698](#)
- obtain a list of the tasks that are eligible to be handlers [29](#)
- obtain the status of a job [905](#)
- published software services templates [311](#)
- read z/OS data set members [826](#)
- read z/OS UNIX file [855](#)
- receive messages from a TSO/E address space [742](#)
- receive messages from applications running in a TSO/E address space [743](#)
- retrieve a list of groups [692](#)
- retrieve a list of portable software instances [507](#)
- retrieve a list of software instances [461](#)

GET method (*continued*)

- retrieve a list of sysplexes [697](#)
- retrieve a list of systems [689](#)
- retrieve a list of systems in a central processor complex (CPC) [701](#)
- retrieve a list of systems in a group [694](#)
- retrieve a list of systems in a sysplex [698](#)
- retrieve a workflow definition [1065](#)
- retrieve data from a remote z/OSMF instance [413](#)
- retrieve data from a secondary z/OSMF instance [413](#)
- retrieve data from an application server [36](#)
- retrieve data from one system [413](#)
- retrieve information about z/OSMF [1003](#)
- retrieve messages from logs [795](#)
- retrieve persisted user or application data [403](#)
- retrieve persisted z/OSMF settings data [1009](#)
- retrieve the contents of a job spool file [914](#)
- retrieve the properties of portable software instances [520](#)
- retrieve the properties of software instances [464](#)
- retrieving data from all the system in a central processor complex (CPC) [413](#)
- retrieving data from all the system in a group [413](#)
- retrieving data from all the system in a sysplex [413](#)
- retrieving data from multiple systems [413](#)
- software services instance names [394](#)
- software services templates [228](#), [244](#)
- storage class definition [621](#)
- storage group activation result [638](#)
- storage group definition [560](#)
- view the properties of portable software instances [520](#)
- view the properties of software instances [464](#)
- volume definition [576](#)
- get source information
 - templates
 - REST API [251](#)

H

handler

- obtaining a list [29](#), [30](#)
- registering [26](#)
- unregistering [31](#)

I

IBM Z and LinuxOne Community

- web site [lix](#)

IBM z/OS Management

- Facility
 - overview [lix](#)
 - programming interfaces [1](#)
 - publications [lix](#)
- web site [lix](#)

IDCAMS

- operations [848](#)

IDCAMS Access Methods Services

- IDCAMS Access Methods Services [848](#)

importing plug-ins [1268](#)

instances

- get response for actions
 - REST API [380](#)
- modifying

- instances (*continued*)
 - modifying (*continued*)
 - REST API [221](#)
 - perform actions
 - REST API [371](#)
 - update variables
 - REST API [364](#)
- isCallable attribute [1111](#), [1137](#)

J

- job
 - cancelling [931](#)
 - changing the job class [928](#)
 - holding [923](#)
 - purging from the JES spool [933](#)
 - releasing [926](#)
 - submitting [918](#)
- job document [936](#), [940](#)
- job file document [940](#)
- job notification document [938](#)
- job spool file
 - retrieving the contents [914](#)
- job spool files list
 - obtaining [912](#)
- JOB statement
 - variable substitution [1130](#)
- job status
 - obtaining [905](#)
- job step data document [941](#)
- jobfiles object [1049](#)
- jobInfo object [1047](#)
- jobs list
 - obtaining [909](#)
- jobstatus object [1048](#)
- Joining the z/OSMF ecosystem
 - developing applications [1225](#)
- JSON document
 - data set list [883](#), [886](#), [889](#)
 - data set list with attributes [884](#)
 - error report document
 - for z/OS data set and file REST interface [893](#)
 - job document [936](#), [940](#)
 - job file document [940](#)
 - job notification document [938](#)
 - job step data document [941](#)
 - UNIX file list [890–892](#)
- JSON document specifications
 - for z/OS data set and file REST interface [883](#)
 - for z/OS jobs REST interface [936](#)
- JSON error report document
 - error reporting [686](#)
 - error reporting categories
 - for storage management API services [639](#)
 - for z/OS data set and file REST interface [893](#)
- JSON web token [1014](#), [1016](#), [1019](#)

K

- keyboard

- keyboard (*continued*)
 - navigation [1305](#)
 - PF keys [1305](#)
 - shortcut keys [1305](#)

L

- list approvals
 - templates
 - REST API [272](#)
- List MVS subsystems
 - List MVS subsystems [436](#)
- List z/OS UNIX Filesystems
 - List z/OS UNIX Filesystems [874](#)
- log in to the z/OSMF server [1014](#)

M

- mainframe education [lix](#)
- messages in logs
 - retrieve [795](#)
- method
 - get the list of JCL JOB statements [999](#)
 - get the list of target systems [1001](#)
- modifying
 - instances
 - REST API [221](#)
- multisystem routing services
 - authenticating with a secondary z/OSMF instance [432](#)
 - authenticating with an HTTP proxy server [434](#)
 - deleting data from a secondary z/OSMF instance [426](#)
 - deleting data from all the systems in a central processor complex (CPC) [426](#)
 - deleting data from all the systems in a group [426](#)
 - deleting data from all the systems in a sysplex [426](#)
 - deleting data from multiple systems [426](#)
 - deleting data from one system [426](#)
 - logging into a remote system [432](#)
 - logging into a secondary z/OSMF instance [432](#)
 - logging into an HTTP proxy server [434](#)
 - overview [408](#)
 - retrieving data from a secondary z/OSMF instance [413](#)
 - retrieving data from all the systems in a central processor complex (CPC) [413](#)
 - retrieving data from all the systems in a group [413](#)
 - retrieving data from all the systems in a sysplex [413](#)
 - retrieving data from multiple systems [413](#)
 - retrieving data from one system [413](#)
 - updating data for a secondary z/OSMF instance [420](#)
 - updating data for all the systems in a central processor complex (CPC) [420](#)
 - updating data for all the systems in a group [420](#)
 - updating data for all the systems in a sysplex [420](#)
 - updating data for multiple systems [420](#)
 - updating data for one system [420](#)
- MVS
 - subsystems [436](#)

N

- navigation
 - keyboard [1305](#)

Notification RESTful Services

description [438](#)

Get all of the notifications received by the current user [439](#)

The notification is sent from a z/OSMF task or z/OSMF user [443](#)

The notification is sent from a z/OSMF task, when the content is the message from the bundle file [441](#)

O

output element [1124](#)

output file

description [1100](#)

P

PATCH

service submission [995](#)

portable software instance [1289](#)

POST

create service submission [982](#)

service submission [997](#)

POST method

add a classification rule [72](#)

Add CFRM policy to administrative policy [676](#)

add portable software instances [525](#)

add software instances [476](#)

add tenant consumer [150](#)

add tenant description [153](#)

add tenant groups [155](#)

approve a software services template [275](#)

approve a software services template, batch [276](#)

Archive a software services template [266](#)

archive a workflow on a z/OS system [1078](#)

assign CPU properties to a tenant [136](#)

assign memory capping properties to a tenant [138](#)

assign solution ID [140](#)

authenticate the user [1014](#)

authenticate with a secondary z/OSMF instance [432](#)

authenticate with an HTTP proxy server [434](#)

batch approve a software services template [276](#)

Copy existing policy with new name [684](#)

create a new software services template based on an existing one [218](#)

create a new version of a software services template [211](#)

create a software services instance [321](#)

create a software services template on a z/OS system [203](#)

Create a UNIX file or directory [862](#)

create a workflow on a z/OS system [1023](#)

create portable software instances [525](#)

create software instances [476](#)

create software service instance names [391](#)

create system variables [803](#)

create variable names [396](#)

create z/OS data sets [837](#)

Create z/OS UNIX zFS Filesystem [876](#)

define volume list storage groups [628](#), [634](#), [636](#)

delete WLM resource pool [749](#)

deploy a software instance [509](#)

disable CPU capping [141](#)

POST method *(continued)*

disable memory capping [143](#)

disable metering [144](#)

edit data for an application server [40](#)

enable CPU capping [146](#)

enable memory capping [147](#)

enable metering [149](#)

Export a defined software instance [482](#)

export system variables [810](#)

import system variables [809](#)

list software instance data sets [470](#)

log into a remote system [432](#)

log into a secondary z/OSMF instance [432](#)

log into an HTTP proxy server [434](#)

missing critical updates [531](#)

missing FIXCAT updates [536](#)

modify a software services instance [221](#)

modify data for an application server [40](#)

obtain a port from a resource pool [61](#)

obtain a SNA application name from a resource pool [67](#)

obtain an IP address from a resource pool [52](#)

perform actions for a software services instance [364](#), [371](#)

POST a notification [441](#), [443](#)

post the request for compliance facts [762](#)

prime WLM resource pool [747](#)

Provision resources defined in an HTTP request body [455](#)

publish a software services template [259](#)

reconnect to a TSO/E address space [718](#)

refresh files for software services templates [265](#)

register an event handler [26](#)

register an event type [24](#)

rejecting an approval for a software services template [278](#)

release a port from a resource pool [65](#)

release a SNA application name from a resource pool [70](#)

release an IP address from a resource pool [58](#)

remove a classification rule [74](#)

remove tenant consumer [152](#)

remove tenant groups [156](#)

retrieve the z/OS system UUID [544](#)

Software Update Search [540](#)

start a TSO/E address space [718](#)

start an application in a TSO/E address space [722](#)

test a software services template [227](#), [261](#)

test a software services template on a z/OS system [268](#)

update data for a remote z/OSMF instance [420](#)

update data for a secondary z/OSMF instance [420](#)

update data for an application server [40](#)

update data for one system [420](#)

update security state [192](#)

update system variables [803](#)

updating data for all the system in a central processor complex (CPC) [420](#)

updating data for all the system in a group [420](#)

updating data for all the system in a sysplex [420](#)

updating data for multiple systems [420](#)

Validate security resources defined in a post body [447](#)

Validate security resources defined in security descriptor file [451](#)

product information files [1285](#)

programming interface

for z/OSMF [1](#)

property file [1268](#)

publish

templates

REST API [259](#)

PUT method

Activate CFRM policy [667](#)

Cancel a job [931](#)

cancel a workflow on a z/OS system [1061](#)

change a user password or passphrase [1016](#)

change the class of a job [928](#)

construct a WLM service definition [751](#), [753](#)

create a tenant [117](#)

create an LPAR resource pool entry [78](#)

delete an LPAR resource pool entry [86](#)

edit data for an application server [40](#)

edit software instances [488](#)

holding a job [923](#)

IDCAMS Access Methods Services [848](#)

issue a command [765](#)

modify an LPAR resource pool entry [82](#)

Modify CFRM policy properties [670](#)

modify data for an application server [40](#)

modify software instances [488](#)

mount file system [879](#), [881](#)

obtain an LPAR resource pool entry [95](#)

persist user or application data [401](#)

persist user or plug-in z/OSMF settings data [1008](#)

ping a TSO/E address space [740](#)

release an LPAR resource pool entry [100](#)

releasing a job [926](#)

Rename CFRM policy [668](#)

retrieve products, features, and FMIDs [495](#)

send messages to a TSO/E address space [737](#)

send messages to an application running in a TSO/E address space [739](#)

start a TSO/E address space [725](#)

start a workflow on a z/OS system [1056](#)

submit a job [918](#)

update data for a remote z/OSMF instance [420](#)

update data for a secondary z/OSMF instance [420](#)

update data for an application server [40](#)

update data for one system [420](#)

update variables for a software services instance [359](#)

updating data for all the system in a central processor complex (CPC) [420](#)

updating data for all the system in a group [420](#)

updating data for all the system in a sysplex [420](#)

updating data for multiple systems [420](#)

write to a z/OS data set or member [831](#)

write to a z/OS UNIX file [859](#)

z/OS data set and PDS member utilities [844](#)

z/OS UNIX file Utilities [867](#)

R

reject approval

templates

REST API [278](#)

Representational State Transfer (REST) interfaces

information resources [1](#)

introduction [1](#)

Resource Measurement Facility

overview [754](#)

run as user element [1149](#)

S

scope attribute [1111](#), [1137](#)

security configuration assistant services

overview [446](#)

sending to IBM

reader comments [lx](#)

ServerPac order

considerations [lix](#)

shortcut keys [1305](#)

software management services

adding portable software instances [525](#)

adding software instances [476](#)

creating portable software instances [525](#)

creating software instances [476](#)

Delete Deployment [519](#)

delete temporary catalog aliases [502](#)

deleting portable software instances [530](#)

deleting software instances [501](#)

deploy a software instance [509](#)

editing software instances [488](#)

listing software instance data sets [470](#)

missing critical updates [531](#)

missing FIXCAT updates [536](#)

modifying software instances [488](#)

obtaining a list of portable software instances [507](#)

obtaining a list of software instances [461](#)

overview [458](#)

removing portable software instances [530](#)

removing software instances [501](#)

retrieve the z/OS system UUID [544](#)

retrieving a list of portable software instances [507](#)

retrieving a list of software instances [461](#)

retrieving products, features, and FMIDs [495](#)

retrieving the properties of portable software instances [520](#)

retrieving the properties of software instances [464](#)

Software Update Search [540](#)

viewing the properties of portable software instances [520](#)

viewing the properties of software instances [464](#)

software management workflows [1297](#)

software service instance name services [390](#)

software services instance services

delete software services instances [367](#)

get [334](#)

get responses for actions [385](#)

get software services instances [345](#)

get variables [354](#), [357](#)

update variables [359](#)

software services template services

delete approval [279](#)

refresh template files [265](#)

step element

approver sub-element [1149](#)

autoEnable attribute [1143](#)

condition attribute [1137](#), [1147](#)

feedback attribute [1115](#)

run as user sub-element [1149](#)

suspend element [1143](#)

step-info object [1039](#), [1087](#)

storage class

definition [621](#)

storage groups

- storage groups (*continued*)
 - activation result [638](#)
 - define volume list [628](#), [634](#), [636](#)
 - definition [560](#)
 - list [549](#)
- storage management
 - data class
 - definition [597](#)
 - storage class
 - definition [621](#)
 - storage groups
 - activation result [638](#)
 - define volume list [628](#), [634](#), [636](#)
 - definition [560](#)
 - list [549](#)
 - volumes
 - definition [576](#)
 - list [571](#)
- subsystem services
 - description [435](#)
- summary of changes
 - z/OSMF Programming Guide [lxiii](#), [lxv](#)
- sysoutDD attribute [1124](#)

T

- templates
 - approving
 - REST API [275](#), [276](#)
 - archiving
 - REST API [266](#)
 - creating new versions
 - REST API [211](#)
 - deleting
 - REST API [227](#)
 - get approval
 - REST API [269](#)
 - get history of
 - REST API [244](#)
 - get properties of
 - REST API [228](#)
 - get source information
 - REST API [251](#)
 - list approvals
 - REST API [272](#)
 - publish
 - REST API [259](#)
 - reject approval
 - REST API [278](#)
 - test running
 - REST API [261](#)
- test running
 - templates
 - REST API [261](#)
- topology services
 - obtaining a list of groups [692](#)
 - obtaining a list of sysplexes [697](#)
 - obtaining a list of systems [689](#)
 - obtaining a list of systems in a central processor complex (CPC) [701](#)
 - obtaining a list of systems in a group [694](#)
 - obtaining a list of systems in a sysplex [698](#)
 - overview [687](#)

- topology services (*continued*)
 - retrieving a list of groups [692](#)
 - retrieving a list of sysplexes [697](#)
 - retrieving a list of systems [689](#)
 - retrieving a list of systems in a central processor complex (CPC) [701](#)
 - retrieving a list of systems in a group [694](#)
 - retrieving a list of systems in a sysplex [698](#)
- trademarks [1310](#)
- TSO/E address space services
 - ending a TSO/E address space [744](#)
 - overview [714](#)
 - pinging a TSO/E address space [740](#)
 - placing a TSO/E address space into dormant state [744](#)
 - receiving messages from a TSO/E address space [742](#)
 - receiving messages from applications running in a TSO/E address space [743](#)
 - sending messages to a TSO/E address space [737](#)
 - sending messages to an application running in a TSO/E address space [739](#)
 - starting a TSO/E address space [718](#)
 - starting an application in a TSO/E address space [722](#)

U

- UNIX file
 - listing [850](#)
 - read [855](#)
 - writing to [859](#)
- UNIX File
 - deleting [865](#)
- UNIX file list [890–892](#)
- UNIX System Services
 - file or directory [867](#)
- user interface
 - ISPF [1305](#)
 - TSO/E [1305](#)
- Utilities
 - operations [844](#), [867](#)

V

- Validate the status of the system [703](#)
- variable
 - input file [1170](#)
 - internal [1163](#)
- variable substitution in the JOB statement [1130](#)
- variable value element [1117](#)
- variable-info object [1050](#), [1095](#)
- variable-reference object [1049](#), [1094](#)
- variables
 - update for instances
 - REST API [364](#)
- versions
 - of templates
 - REST API [211](#)
- volumes
 - definition [576](#)
 - list [571](#)

W

- WLM

- WLM (*continued*)
 - resource pool [747, 749](#)
 - service definition [751, 753](#)
- WLM resource pool
 - delete [749](#)
 - prime [747](#)
- WLM service definition
 - construct [751, 753](#)
- workflow
 - terms [1100](#)
 - upgradeable [1113](#)
- workflow internal variable
 - description [1163](#)
- workflow variable input file
 - creating [1170](#)
 - description [1170](#)
 - variable substitution [1171](#)

Z

- z/OS Basic Skills information center [lix](#)
- z/OS compliance REST
 - interface
 - overview [760](#)
- z/OS Compliance REST Interface
 - post the request for compliance facts [762](#)
- z/OS console REST interface
 - description [547, 658, 746, 763](#)
- z/OS data set and PDS member utilities
 - z/OS data set and PDS member utilities [844](#)
- z/OS data sets and files REST interface
 - Create a UNIX file or directory [862](#)
 - creating z/OS data sets [837](#)
 - delete a PDS or PDSE [842](#)
 - delete a UNIX file or directory [865](#)
 - deleting z/OS data sets [840](#)
 - description [813](#)
 - listing UNIX files and directories [850](#)
 - listing z/OS data set members [823](#)
 - listing z/OS data sets [820](#)
 - mount a unix file system [881](#)
 - mount a UNIX file system [879](#)
 - reading z/OS UNIX files [855](#)
 - retrieving z/OS data set members [826](#)
 - writing to data set members [831](#)
 - writing to z/OS UNIX files [859](#)
 - z/OS UNIX file Utilities [867](#)
- z/OS jobs REST interface
 - cancelling a job on z/OS [931](#)
 - changing the class of a job [928](#)
 - description [900](#)
 - holding a job [923](#)
 - listing the jobs for an owner, prefix, or job ID [909](#)
 - listing the spool files for a job [912](#)
 - obtaining the status of a job [905](#)
 - purging a job from the JES spool [933](#)
 - releasing a job [926](#)
 - retrieving messages from logs on the system [795](#)
 - retrieving the contents of a job spool file [914](#)
 - submitting a job to run on z/OS [918](#)
- z/OS Management Services Catalog [953](#)

- z/OS Management Services Catalog
 - Services
 - get the list of JCL JOB statements [999](#)
 - get the list of target systems [1001](#)
- z/OS system variable
 - services
 - overview [802](#)
- z/OS UNIX file
 - write [859](#)
- z/OS variable services
 - overview [802](#)
- z/OSMF authentication
 - services
 - description [1013](#)
- z/OSMF cloud provisioning resource management
 - services
 - add tenant consumer [150](#)
 - add tenant description [153](#)
 - add tenant groups [155](#)
 - assign CPU properties to a tenant [136](#)
 - assign memory capping properties to a tenant [138](#)
 - assign solution ID [140](#)
 - delete a tenant [135](#)
 - disable CPU capping [141](#)
 - disable memory capping [143](#)
 - disable metering [144](#)
 - enable CPU capping [146](#)
 - enable memory capping [147](#)
 - enable metering [149](#)
 - remove tenant consumer [152](#)
 - remove tenant groups [156](#)
 - update security state [192](#)
- z/OSMF cloud provisioning resource pool services
 - add a classification rule [72](#)
 - obtain a port from a resource pool [61](#)
 - obtain a SNA application name from a resource pool [67](#)
 - obtain an IP address from a resource pool [52](#)
 - release a port from a resource pool [65](#)
 - release a SNA application name from a resource pool [70](#)
 - release an IP address from a resource pool [58](#)
 - remove a classification rule [74](#)
- z/OSMF Cloud Provisioning services [102](#)
- z/OSMF Community Guild
 - web site [lix](#)
- z/OSMF information retrieval service
 - overview [1002](#)
 - retrieving information about z/OSMF [1003](#)
- z/OSMF Programming Guide
 - content, changed [lxix](#)
 - content, new [lxiii, lxv](#)
 - content, no longer included [lxix](#)
 - summary of changes [lxiii, lxv](#)
- z/OSMF published software services template
 - services
 - get [290, 302](#)
 - list [311](#)
- z/OSMF resource management services
 - creating a tenant [117](#)
 - creating an LPAR resource pool entry [78](#)
 - deleting an LPAR resource pool entry [86](#)
 - get metered data [755](#)
 - Get the properties of an LPAR entry [91](#)
 - getting a domain [104, 110](#)

- z/OSMF resource management services (*continued*)
 - getting a domain resource pool [164](#)
 - getting a resource pool [158](#)
 - getting a resource pool history [171](#)
 - getting a tenant for cloud provisioning [121](#)
 - getting a tenant history for cloud provisioning [126](#)
 - getting data set attributes [76](#)
 - getting security profile information [194](#)
 - getting z/OSMD tsoAPP information [734](#)
 - issue a TSO/E command with z/OSMF REST API [725](#)
 - list LPAR resource pool entries [87](#)
 - list the domains for cloud provisioning [112](#)
 - listing domain resource pools [179](#)
 - listing resource pools [173](#)
 - listing template resource pools [186](#)
 - listing the tenants for cloud provisioning [129](#)
 - modifying an LPAR resource pool entry [82](#)
 - obtaining an LPAR resource pool entry [95](#)
 - releasing an LPAR resource pool entry [100](#)
- z/OSMF resource measurement
 - services
 - get metered data [758](#)
- z/OSMF security configuration assistant services
 - Provision security resources defined in an HTTP request body [455](#)
 - Validate security resources defined in a post body [447](#)
 - Validate security resources defined in security descriptor file [451](#)
- z/OSMF settings services
 - overview [1006](#)
 - persisting user or plug-in z/OSMF settings data [1008](#)
 - retrieving z/OSMF settings data [1009](#)
- z/OSMF software services instance name
 - services
 - list [394](#)
- z/OSMF software services instances name
 - services
 - create variable names [396](#)
- z/OSMF software services instances
 - services
 - create [321](#)
 - create names [391](#)
 - delete response for action [388](#)
 - get responses for actions [380](#)
 - modify [221](#)
 - perform actions [371](#)
 - update variables [364](#)
- z/OSMF software services template services
 - approve an approval record [275](#), [276](#)
 - Archiving a software services template [266](#)
 - creating a new software services template based on an existing one [218](#)
 - creating a new version of a software services template [211](#)
 - creating a software services template [259](#)
 - creating a software services template on a z/OS system [203](#)
 - deleting a software services template [227](#)
 - get an approval of a software services template [269](#)
 - get prompt variables [306](#)
 - get source information for software services template [251](#)

- z/OSMF software services template services (*continued*)
 - list approvals for a software services template [272](#)
 - rejecting an approval of a software services template [278](#)
 - testing a software services template [261](#)
 - testing a software services template on a z/OS system [268](#)
- z/OSMF software services templates
 - services
 - get [228](#)
 - get history [244](#)
- z/OSMF storage management
 - data class definition [597](#)
 - define volume list storage groups [628](#), [634](#), [636](#)
 - list data classes [580](#)
 - list storage classes [614](#)
 - list storage groups [549](#)
 - list volumes [571](#)
 - storage class definition [621](#)
 - storage group activation result [638](#)
 - storage group definition [560](#)
 - volume definition [576](#)
- z/OSMF system variable services
 - creating system variables [803](#)
 - deleting system variables [812](#)
 - exporting system variables [810](#)
 - getting the system variables [805](#)
 - importing system variables [809](#)
 - updating system variables [803](#)
- z/OSMF workflow services
 - archiving a workflow on a z/OS system [1078](#)
 - cancel a workflow on a z/OS system [1061](#)
 - creating a workflow on a z/OS system [1023](#)
 - delete a workflow on a z/OS system [1063](#)
 - delete an archived workflow on a z/OS system [1096](#)
 - get the properties of a z/OSMF workflow [1031](#)
 - get the properties of an archived workflow [1082](#)
 - listing the archived workflows on a z/OS system [1080](#)
 - listing the workflows on a z/OS system [1054](#)
 - overview [1020](#)
 - retrieving a workflow definition [1065](#)
 - starting a workflow on a z/OS system [1056](#)



Product Number: 5650-ZOS

SC27-8420-50

