

IBM Workload Scheduler for z/OS



Planning and Installation

Version 9.3 SPE (Revised January 2019)

IBM Workload Scheduler for z/OS



Planning and Installation

Version 9.3 SPE (Revised January 2019)

Note

Before using this information and the product it supports, read the information in "Notices" on page 401.

This edition applies to version 9, release 3, modification level 0 of IBM Workload Scheduler for z/OS (program number 5698-T08) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1999, 2016.**

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

© Copyright HCL Technologies Limited 2016, 2019.

Contents

Figures ix

Tables xi

About this publication xiii

What is new in this release xiii

Who should read this publication. xiii

Accessibility xiv

Technical training xiv

Support information xiv

Part 1. Planning 1

Chapter 1. Overview 3

Hardware and software requirements 3

Hardware requirements. 3

Software requirements and optional software 4

Related software 5

Parts and their relationships 5

Tracker 6

Controller 6

Server 7

Graphical user interfaces 7

Data Store 8

Output collector 9

Configurations. 9

Subtasks 11

Relationship between the Scheduler and z/OS. 13

Using the IBM Workload Scheduler for z/OS

Program Directory 14

Sample library 14

The installation process 15

Chapter 2. Planning your configuration 17

Planning considerations 17

Trackers 17

Initialization statements 17

Communication 18

How to connect IBM Workload Scheduler for z/OS

systems. 18

Shared DASD 18

z/OS cross-system coupling facility 19

VTAM (network communication function) 19

TCP/IP. 19

Workstation destination 20

Workload restart. 20

JES considerations 21

Basic server configuration example 21

Basic configuration examples 23

DASD connected 23

VTAM connected 24

TCP/IP connected 26

XCF connected 27

Tracker and controller in a single address space 29

Basic data store configuration examples 30

SNA only connection 30

XCF only connection 31

TCP/IP only connection 33

Mixed SNA and XCF connection 35

Mixed TCP/IP and XCF connection 37

Chapter 3. Planning your installation 41

Installation considerations 41

Configuring for availability 41

Hot standby 41

Starting an event writer with an event reader

function 42

Using a Hierarchical File System cluster 42

Using two message log (MLOG) data sets 42

Checklist for installing IBM Workload Scheduler for z/OS 44

Part 2. Installing IBM Workload Scheduler for z/OS 51

Chapter 4. Installing 53

Step 1. Loading tracker software 55

Step 2. Loading controller software 55

Step 3. Loading national language support software 56

Step 4. Using the EQQJOBS installation aid. 56

Setting up the EQQJOBS installation aid. 57

Creating the sample job JCL. 58

Generating batch-job skeletons 69

Generating Data Store samples 74

Creating the Workload Automation Programming

Language samples 78

Step 5. Adding SMF and JES exits for event tracking 81

SMF only 82

JES2 only 82

JES3 only 83

Step 6. Updating SYS1.PARMLIB 83

Defining subsystems 83

Authorizing the load-module library 86

Updating SMF parameters 86

Updating z/OS dump options 88

Updating the z/OS link-library definition 88

Updating XCF initialization options 89

Modifying TSO parameters 90

Performance considerations 91

Defining the DLF exit for Hiperbatch support 91

Starting the product automatically. 91

Updating APPC options 91

Implementing support for data set triggering 92

Step 7. Setting up the RACF environment 93

Controlling the user ID of the address space 93

Controlling the user ID of submitted jobs 93

Protecting data sets. 94

Controlling access to resources 95

Controlling access to IBM Workload Scheduler for z/OS resources when using the Dynamic Workload Console	96
Authorizing IBM Workload Scheduler for z/OS as a job submitter	97
Authorizing IBM Workload Scheduler for z/OS to issue JES commands	97
Authorizing IBM Workload Scheduler for z/OS E2E server task to create USS processes	98
Authorizing IBM Workload Scheduler for z/OS E2E and Dynamic Workload Console server tasks for security resource EZB.BINDDVIPARANGE	99
Step 8. Securing communications	99
Security for TCP/IP connections	99
Security for HTTP connections	101
Step 9. Allocating data sets	102
Allocating the VSAM data sets	102
Allocating Restart and Cleanup VSAM data sets	108
Allocating non-VSAM data sets	109
Allocating Data Store data sets	123
Allocating data sets for the Dynamic Workload Console reporting feature	124
Allocating the files and directories	124
Step 10. Creating JCL procedures for address spaces	128
Implementing support for started-task operations	129
Required data sets	130
Optional data sets	132
Step 11. Defining the initialization statements.	134
Step 12. Creating the DB2 database	134
Sample to migrate the history database.	135
Step 13. Setting up the ISPF environment	136
Setting up the CLIST library	136
Setting up the ISPF tables	136
Allocating dialog data sets to your TSO session	139
Invoking the IBM Workload Scheduler for z/OS dialog	140
Step 14. Using XCF for communication.	141
XCF groups	141
XCF runtime options	142
Initialization statements used for XCF	143
Step 15. Activating the network communication function	143
Adding NCF to the VTAM network definitions	144
Adding NCF session parameters	145
Activating network resources	146
Diagnostic data set	146
Step 16. Using TCP/IP for communication	146
Initialization statements used for TCP/IP	147
Step 17. Activating support for the API.	147
Defining VTAM resources	148
Updating APPC options	149
Activating support for APPC	150
Step 18. Activating support for the product dialog and programming interface using the server	150
Defining VTAM resources for the product dialog and program interface using the server	151
Defining VTAM resources for the server	151
Updating APPC options for the server	153
Defining VTAM resources in a parallel sysplex	153

Starting the server	154
Step 19. Activating support for the end-to-end scheduling with fault tolerance capabilities	154
Activating server support for the end-to-end scheduling with fault tolerance capabilities	154
Step 20. Activating support for end-to-end scheduling with z-centric capabilities	155
Step 21. Activating support for Dynamic Workload Console	155
Prerequisites.	155
Considerations	156
Activating server support for the Dynamic Workload Console.	157
Step 22. Activating support for the Java utilities	157
Step 23. Activating support for FIPS standard over SSL secured connections.	158

Chapter 5. Verifying your installation 161

Overview of verification.	161
Verifying installation of a tracker	161
Ensuring that all installation tasks are complete	162
Checking the message log (EQQMLOG)	162
Verifying tracking events	163
Performing problem determination for tracking events	164
Verifying installation of a controller and dialogs	167
Ensuring that all installation tasks are complete	167
Checking the message log (EQQMLOG)	168
Checking the server message log	168
Checking dialog functions	169
Performing problem determination	169
Verifying installation of a standby controller	170
Ensuring that all installation tasks are complete	171
Checking the message log (EQQMLOG)	171
Verifying installation of the Restart and Cleanup function	172
Checking the message log (EQQMLOG)	173
Verifying configuration	174
Creating entries in the databases	174
Running batch jobs	174
Checking the message logs (EQQMLOG)	174
Verifying workload submission	182
Verifying job submission	183
Verifying takeover by a standby controller.	183

Chapter 6. Migrating 185

Planning for migration	185
Migration considerations	185
Customization considerations	186
Migration strategies	186
Migration overview	189
Installation and verification.	189
Parallel testing	190
Migrating an end-to-end network	191
Migrating DB2 history	191
Migrating DB2 reporting	191
Migrating the backup controller	192
Changing a shared DASD tracker-to-controller connection to an NCF, XCF, or TCP/IP connection	192

Running on upgraded operating systems	194
Migrating actions	194
Migrating data sets	194
Migrating the production environment	201
Migrating the controller with the	
IWSZSELFUPGRADE application	203
Installing or upgrading the IBM Workload	
Scheduler for z/OS agent automatically	205
Performing fallback	207

Part 3. IBM Workload Scheduler for z/OS connector 211

Chapter 7. Installing, upgrading, and uninstalling IBM Workload Scheduler for z/OS connector on distributed systems. 213

IBM Workload Scheduler for z/OS DVDs to use	213
Upgrading IBM Workload Scheduler for z/OS connector single instance on distributed systems	213
Upgrading z/OS connector installed with one or more components in the same directory	214
Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector	215
Upgrading the fault-tolerant agent and the z/OS connector	216
Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector.	217
Upgrading the dynamic domain manager and the z/OS connector	218
Procedure to upgrade the Dynamic Workload Console and the z/OS connector	219
Managing z/OS engines by using WebSphere Application Server tools	219
Creating a z/OS connector instance	220
Displaying the properties of a z/OS connector instance	221
Updating the properties of a z/OS connector instance	222
Listing all the z/OS connector instances	224
Removing a z/OS connector instance	224
Configuring the Dynamic Workload Console to work with a remote z/OS connector.	225
Encrypting the connection between the z/OS connector and the server started task	231

Chapter 8. Installing and uninstalling on WebSphere Application Server for z/OS 233

Business Scenario	233
Authorization roles required for installing and uninstalling	233
Installing on WebSphere Application Server for z/OS	233
Installing using the Integrated Solution Console	234
Installing using the zConnInstall.sh script	235
Installation and uninstallation log files	237
Install the additional plug-in	238

Enabling communications with Dynamic Workload Console	238
Applying maintenance	241
Uninstalling	242
Uninstalling using the Integrated Solution Console	242
Uninstalling using the zConnUninstall.sh script	242
Uninstalling additional plug-ins	243

Chapter 9. Troubleshooting and maintaining the installation 245

Maintaining the installation of the z/OS connector	245
Updating the SOAP properties after changing the WebSphere Application Server user or its password.	245
Updating the SOAP properties usage	245

Part 4. Dynamic Workload Console 247

Chapter 10. Overview of the Dynamic Workload Console 249

Chapter 11. Preparing 251

Directories created outside of TWA_home at installation time	251
Accessing the installation media	252
Using DVDs.	252
Downloading images on your workstation	253
Creating a network repository	253

Chapter 12. Dynamic Workload Console prerequisites. 255

Supported operating systems	255
Scanning system prerequisites for IBM Workload Scheduler.	256
Scanning system prerequisites for Installation Manager	257

Chapter 13. Installing 259

Selecting your installation method	259
Installation procedure for Dynamic Workload Console	259
WebSphere Application Server profile configuration	263
z/OS connector configuration	264
Performing a silent installation	264
Performing a Dynamic Workload Console silent installation	264
Performing a Dynamic Workload Console and its prerequisites silent installation	266
Encrypting user passwords for response files	268
Installation log files	270
Interactive wizard installation and uninstallation log files	270
Jazz for Service Management extension for WebSphere profile log files	272
Accessing the Dynamic Workload Console	272
Quick steps to define an IBM Workload Scheduler engine connection	274

Quick steps to define an dynamic workload broker connection	274
Starting and stopping the Dynamic Workload Console	275
Chapter 14. Configuring	277
Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP).	278
Chapter 15. Navigating the Dynamic Workload Console	279
First actions	279
Chapter 16. Upgrading	281
Upgrading overview	281
Upgrading a Dynamic Workload Console V8.6 single instance	282
Upgrading a Dynamic Workload Console V8.6 instance on the same workstation.	282
Upgrading Dynamic Workload Console V8.5.1 single instance on a new workstation	285
Upgrading Dynamic Workload Console V8.6 single instance on a new workstation	288
Upgrading Dynamic Workload Console V8.6 installed with one or more components in the same directory	292
Upgrading the fault-tolerant agent and the Dynamic Workload Console	293
Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector	294
Upgrading the dynamic domain manager and the Dynamic Workload Console	294
Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector.	295
Procedure to upgrade the Dynamic Workload Console and the z/OS connector	296
Upgrading Dynamic Workload Console V9.1 or later	296
Chapter 17. Uninstalling	299
Uninstalling using the Installation Manager wizard	299
Uninstalling in silent mode.	299
Chapter 18. Troubleshooting the installation, upgrade, and uninstallation	301
Installation and uninstallation log and trace files	301
Manually uninstall the Dynamic Workload Console and the zConnector on Windows systems	301
Manually uninstall the Dynamic Workload Console and the zConnector on UNIX systems	302
Troubleshooting scenarios	304
Problems with the launchpad	304
Problems with the interactive installation	305
Problems with the silent installation.	305

Part 5. IBM Workload Scheduler for z/OS Agent 307

Chapter 19. Installing the IBM Workload Scheduler for z/OS Agent. 309

User authorization requirements	310
Authorization roles for running the twsinst script	310
Installing using twsinst	310
twsinst	310
The twsinst log files	315
Creating a Docker image to run IBM Workload Scheduler for z/OS Agents	315
Enabling dynamic capabilities after installation or upgrade	316
Installing the additional plug-ins by using the IBM Workload Scheduler for additional plug-ins	316

Chapter 20. Upgrading the IBM Workload Scheduler for z/OS Agent. 317

Coexistence with previous versions	317
User authorization requirements	317
Upgrading notes	317
Upgrading using twsinst	317
Upgrading process	317
Examples.	320
The twsinst log files	320
Enabling dynamic capabilities after upgrade	321

Chapter 21. Uninstalling the IBM Workload Scheduler for z/OS Agent. 323

User authorization requirements	323
Uninstalling the IBM Workload Scheduler for z/OS Agent using the twsinst script.	323
The twsinst log files	324
Uninstalling the additional plug-ins using the IBM Workload Scheduler for additional plug-ins	325

Chapter 22. Troubleshooting and maintaining installation, upgrade, and uninstallation 327

Analyzing return codes for agent installation, upgrade, restore, and uninstallation	327
twsinst needs long time to run if the machine does not have enough temporary space	329

Part 6. IBM Workload Scheduler for z/OS Agent on IBM i systems 331

Chapter 23. Prerequisites 333

Chapter 24. Installing agents on IBM i systems. 335

Agent installation parameters on IBM i systems	337
Example installation of an agent on IBM i systems	340

Chapter 25. Upgrading agents on IBM i systems 341

Agent upgrade parameters on IBM i systems . . . 342
Example upgrade of an agent on IBM i systems . . . 344

Chapter 26. Uninstalling agents on IBM i systems 345

Chapter 27. The twsinst script log files on IBM i systems 347

Chapter 28. Analyzing return codes for agent installation, upgrade, restore, and uninstallation 349

Part 7. Appendixes 353

Appendix A. Sample library (SEQQSAMP) 355

Using the Visual Age compiler 360
SMP/E samples 361
 Environment setup 361
 RECEIVE processing 361
 APPLY processing 362
 ACCEPT processing 363
SMF exits. 363
 Exit installation. 364
 Job step termination exit. 364
 Initialization exit 365
 Record write exits 365
JES exits 365
 Exit installation. 366
 JES2 QMOD phase change exit 366
 JES2 JCT I/O exit 366
 JES3 OSE modification exit 366
 JES3 input service final-user exit 366
RACF samples 366

Class descriptor table. 367
Router table 367
Sample library (SEQQSAMP) 367

Appendix B. Configuration examples 371

The controlling system 371
Automatic restart actions 371
Initialization statements 371
Multi-access spool systems connected through shared DASD 371
Individual systems connected via shared DASD 374
A z/OS Sysplex 376
A PLEX configuration 379
Controlling a z/OS system through a VTAM link 381
Controlling a z/OS system through a TCP/IP link 382
Controlling a JES2 MAS system through a VTAM link. 384

Appendix C. Invoking the EQQEXIT macro 387

Invoking EQQEXIT in SMF exits 387
Invoking EQQEXIT in JES exits 387
 Macro invocation syntax for EQQEXIT 388

Appendix D. Invoking the EQQLSENT macro 391

Invoking EQQLSENT to create EQQDSLST 391
 Macro invocation syntax for EQQLSENT 391

Appendix E. Using response files. . . 395

The Dynamic Workload Console response file properties 395

Notices 401

Trademarks 403
Terms and conditions for product documentation 403

Index 405

Figures

1.	A basic server configuration example	22	20.	EQQJOBS2 - Generate IBM Workload Scheduler for z/OS batch-job skeletons	71
2.	A z/OS system connected through shared DASD	24	21.	EQQJOBSA - Generate IBM Workload Scheduler for z/OS batch-job skeletons	72
3.	A z/OS system with a VTAM connection	25	22.	EQQJOBS5 - Create Data Store samples	75
4.	A z/OS system with a TCP/IP connection	27	23.	EQQJOBS6 - Create Data Store samples	76
5.	A z/OS system with an XCF connection	28	24.	EQQJOBS7 - Create Data Store samples	77
6.	A tracker and controller configured in a single address space	29	25.	EQQJOBSE - Create WAPL samples	79
7.	Controller and tracker in same address space with tracker connected through SNA	30	26.	Sample message log for a standby controller	172
8.	Controller, tracker, and Data Store connected through XCF	32	27.	Shows the keys to enable SSL between the z/OS connector and the Dynamic Workload Console	239
9.	Controller and tracker in same address space with tracker connected through TCP/IP	34	28.	Two z/OS JES2 MAS complexes connected through shared DASD	372
10.	A mixed SNA and XCF connection.	35	29.	Individual systems connected through shared DASD	375
11.	A mixed TCP/IP and XCF connection.	38	30.	A z/OS Sysplex.	377
12.	EQQJOBS0 - EQQJOBS application menu	57	31.	An IBM Workload Scheduler for z/OS PLEX environment	380
13.	EQQJOBS3 - Create sample job JCL	58	32.	Controlling a z/OS system through a VTAM link	381
14.	EQQJOBS4 - Create sample job JCL	59	33.	Controlling a z/OS system through a TCP/IP link	383
15.	EQQJOBS8 - Create sample job JCL	61	34.	Controlling a JES2 MAS system through a VTAM link	385
16.	EQQJOBS9 - Create sample job JCL	63			
17.	EQQJOBSC - Create sample job JCL	64			
18.	EQQJOBSD - Create sample job JCL	66			
19.	EQQJOBS1 - Generate IBM Workload Scheduler for z/OS batch-job skeletons	70			

Tables

1.	IBM Workload Scheduler for z/OS subtasks	11	37.	Upgrade deployment model for Dynamic Workload Console multiple instances installed in the same directory	215
2.	Stages summarizing the IBM Workload Scheduler for z/OS installation process	15	38.	Configuration page settings.	234
3.	Example EQQSERP members for Figure 1	22	39.	zConnInstall.properties properties and corresponding values	236
4.	Example EQQPARM members for Figure 2	24	40.	zConnUpdate.properties properties and corresponding values	241
5.	Example EQQPARM members for Figure 3	26	41.	zConnUninstall.properties properties and corresponding values	243
6.	Example EQQPARM members for Figure 4	27	42.	Dynamic Workload Console required elmages	253
7.	Example EQQPARM members for Figure 5	28	43.	Installation response files	266
8.	Example EQQPARM members for Figure 6	29	44.	Upgrade response files	266
9.	Example members for Figure 7	31	45.	Global installation response files	268
10.	Example members for Figure 8	32	46.	Upgrade single or multiple instances.	281
11.	Example members for Figure 9	34	47.	Upgrade deployment model for Dynamic Workload Console multiple instances in the same directory	293
12.	Example members for Figure 10.	36	48.	Uninstallation response files	300
13.	Example members for Figure 11.	38	49.	Required authorization roles for running twsinst.	310
14.	Checklist for installing IBM Workload Scheduler for z/OS	44	50.	Windows operating system agent return codes	327
15.	IBM Workload Scheduler for z/OS installation tasks	53	51.	UNIX or Linux operating system agent return codes	328
16.	tracker libraries loaded by SMP/E	55	52.	Valid values for -lang and LANG parameter	338
17.	Controller libraries loaded by SMP/E.	55	53.	Windows operating system agent return codes	349
18.	NLS libraries loaded by SMP/E.	56	54.	UNIX or Linux operating system agent return codes	350
19.	Sample JCL generated by the EQQJOBS dialog	67	55.	SEQQSAMP library members	355
20.	Controller skeleton JCL generated by the EQQJOBS dialog	73	56.	Example EQQPARM members for the previous figure	373
21.	Data Store samples generated by the EQQJOBS dialog	78	57.	Example EQQPARM members for the previous figure	376
22.	Workload Automation Programming Language samples generated by the EQQJOBS dialog	80	58.	Example EQQPARM members for the previous figure	378
23.	Sample exits for IBM Workload Scheduler for z/OS.	81	59.	Example EQQPARM Members for the previous figure	380
24.	Examples of MAXECSA storage values	85	60.	Example EQQPARM Members for the previous figure	382
25.	IBM Workload Scheduler for z/OS VSAM data sets	102	61.	Example EQQPARM Members for the previous figure	383
26.	Calculations of VSAM data set size	105	62.	Example EQQPARM members for the preceding figure	386
27.	IBM Workload Scheduler for z/OS non-VSAM data sets	109	63.	General information	395
28.	Data Store VSAM data sets	124	64.	WebSphere configuration	396
29.	Started task JCL samples for IBM Workload Scheduler for z/OS address spaces	128	65.	z/OS connector configuration	397
30.	IBM Workload Scheduler for z/OS required data sets	130	66.	Offering and features to install.	398
31.	IBM Workload Scheduler for z/OS optional data sets	132	67.	Upgrade settings	398
32.	ISPF and IBM Workload Scheduler for z/OS dialog data sets	139			
33.	Problem determination for missing tracking events	164			
34.	Data sets that you need to migrate	198			
35.	Data sets that IBM Workload Scheduler for z/OS can use	199			
36.	IBM Workload Scheduler for z/OS bundles	213			

About this publication

IBM Workload Scheduler for z/OS: Planning and Installation describes the configuration planning and installation tasks of IBM Workload Scheduler for z/OS. Installation is the task of making a program ready to do useful work. This task includes adding the materials on the IBM® distribution tape to your system, initializing the program, and applying PTFs to the program. When you install a product, you are carrying out decisions you made in the planning step. Customization, an optional step, gives you the opportunity to tailor the program to the behavior or special needs required for your site.

Your workload can run on various platforms, but you control it from a central z/OS® system that runs the IBM Workload Scheduler for z/OS controller.

The term *scheduler*, when used in this publication, refers to IBM Workload Scheduler for z/OS. The term *DB2®*, when used in this publication, refers to both DATABASE 2 and DB2 Universal Database™.

The term *z/OS* is used in this publication to mean z/OS and OS/390® operating systems. Where the term *OS/390* appears, the related information applies only to OS/390 operating systems.

This publication complements the *IBM Workload Scheduler for z/OS Program Directory* that describes how to add the materials on the IBM distribution tape to your system.

The *Program Directory* is provided with the IBM Workload Scheduler for z/OS installation tape. It describes all of the installation materials and gives installation instructions specific to the product release level or feature number. If any differences exist between this publication and the *Program Directory*, use the information in the *Program Directory*.

What is new in this release

Learn what is new in this release.

For information about the new or changed functions in this release, see *Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, see the Program Directory and the Dynamic Workload Console Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045183>.

Who should read this publication

Learn the audience of this publication.

This publication is intended for system programmers who are responsible for software on a z/OS system and plan on installing IBM Workload Scheduler for z/OS.

To use this publication effectively, you must be familiar with the following topics:

- Job control language (JCL)
- IBM System Modification Program Extended (SMP/E)
- z/OS
- JES concepts and facilities
- Writing small code fragments in the Assembler H language
- Interactive System Productivity Facility (ISPF)
- Interactive System Productivity Facility/Program Development Facility (ISPF/PDF)
- Time-Sharing Option (TSO)
- Virtual Storage Access Method (VSAM) (desirable but not essential)

The IBM Workload Scheduler for z/OS Application Programming Interface (API) uses advanced program-to-program communication (APPC) services. Defining and configuring the conversation partners requires some knowledge of APPC services.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information, see the Accessibility Appendix in the *IBM Workload Scheduler User's Guide and Reference*.

Technical training

Cloud & Smarter Infrastructure provides technical training.

For Cloud & Smarter Infrastructure technical training information, see:
<http://www.ibm.com/software/tivoli/education>

Support information

IBM provides several ways for you to obtain support when you encounter a problem.

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix about support information in *IBM Workload Scheduler: Troubleshooting Guide*.

Part 1. Planning

This part provides an overview of the IBM Workload Automation environment and describes how to plan for the installation.

Chapter 1. Overview

An overview of IBM Workload Scheduler for z/OS.

About this task

You can use IBM Workload Scheduler for z/OS to plan, control, and automate the entire production workload in your complex, not just the z/OS batch subset. It automatically plans, controls, and monitors your production workload to maximize the throughput and optimize resource use, but lets you intervene manually when required. If you are currently using previously supported versions, follow the instructions in Chapter 6, “Migrating,” on page 185.

This chapter introduces IBM Workload Scheduler for z/OS and its implementation.

If you are not familiar with IBM Workload Scheduler for z/OS terminology or functions, read the *Tivoli Workload Automation: Overview*.

Hardware and software requirements

Hardware and software requirements of IBM Workload Scheduler for z/OS.

About this task

Hardware and software requirements of IBM Workload Scheduler for z/OS and includes optional and related software.

Hardware requirements

About this task

IBM Workload Scheduler for z/OS operates on any IBM hardware configuration supported by z/OS Version 1.12 (program number 5694-A01) or later.

IBM Workload Scheduler for z/OS needs a minimum region of 8 MB below the 16 MB line; at least 32 MB must be available above the 16 MB line. The region value depends strictly on the IBM Workload Scheduler for z/OS customization and workload. For IBM Workload Scheduler for z/OS to work correctly, you might need to specify a region value of 64 MB, which gives you all the available space below the 16 MB line plus 64 MB above the 16 MB line.

In particular, to avoid storage problems, the IBM Workload Scheduler for z/OS server must run with a region value of 64 MB when you use the end-to-end scheduling with fault tolerance capabilities. In addition, a region value of 64 MB is strongly recommended when the IBM Workload Scheduler for z/OS TCP/IP or APPC server is used for the remote interfaces, for example, the Dynamic Workload Console, PIF, and ISPF.

Consider to increase the region size specification if the server task will normally run for a long time (several weeks or months). Also make sure that the IEFUSI exit is not limiting the region size to a value less than the one coded in the JCL.

An IBM Workload Scheduler for z/OS dialog user needs a region of 3 MB below the 16 MB line; if you want to run EQQAUDIT interactively (option 9.10 of the main menu), this number then increases to 4 MB. IBM Workload Scheduler for z/OS report programs need a region of 3 MB below the 16 MB line.

IBM Workload Scheduler for z/OS uses less than 1 KB of 24-bit Common Service Area (CSA) storage. The amount of 31-bit Extended Common Service Area (ECSA) used is approximately 30 KB, plus 2 KB per active dialog user.

A display terminal supported by ISPF Version 6.1 or later is required to invoke and run the IBM Workload Scheduler for z/OS host dialogs.

The graphic function requires a terminal supporting Graphic Data Display Manager (GDDM/MVS) Version 3 Release 2 or later.

Software requirements and optional software

About this task

Installing and maintaining IBM Workload Scheduler for z/OS requires one of the following:

- z/OS (5694-A01) Version 1.13, or later.
- IBM SMP/E for z/OS Version 3 Release 5 (program number 5655-G44), or later.

IBM Workload Scheduler for z/OS requires the functions provided by a z/OS control program running on a z/OS system. The Job Entry Subsystem might be either JES2 or JES3.

Controlling system

About this task

The following IBM licensed programs are required on the IBM Workload Scheduler for z/OS controlling system:

- IBM Workload Scheduler for z/OS Version 9.3 (program number 5698-T08). Both the base product (the tracker) and the controller feature are required.

Controlled z/OS systems

About this task

On each z/OS system that is controlled by IBM Workload Scheduler for z/OS, one of the following IBM licensed programs is required:

- IBM Workload Scheduler for z/OS (program number 5698-T08). Only the base product (the tracker) is required.

Optional software

About this task

The following IBM Workload Scheduler for z/OS functions require specific IBM programs:

- Tracking resource availability requires the Resource Object Data Manager (RODM) in IBM Tivoli NetView for z/OS (program number 5697-NV6).
- Graphical view of jobs and their dependencies using ISPF panels requires GDDM.
- For TCP/IP Communications, z/OS V1R13.0 (or later) Communications Server is required.

- IBM Tivoli NetView for z/OS (program number 5697-NV6) is required to enable IBM Workload Scheduler for z/OS to schedule generic alerts as defined by that NetView[®] release and to specify an alert receiver ID other than the default receiver.
- User-authority-support functions require z/OS V1R13.0 (or later) Security Server RACF[®] (5694-A01).
- z/OS V1R13.0 DFSMS (5694-A01) with Hierarchical Storage Management component is required for the catalog management function to recall migrated data sets.
- IBM DB2 for z/OS V10, or later (program number 5605-DB2) is required for IBM Workload Scheduler for z/OS history functions and Dynamic Workload Console reporting feature.
- The IBM Workload Scheduler for z/OS Control Language tool and the Dynamic Workload Console reporting feature require either the IBM Compiler Library for REXX/370 (program number 5695-014) or the IBM Alternate Library for REXX on zSeries, which can be downloaded from <http://www-01.ibm.com/support/docview.wss?rs=960&uid=swg24006107>.
- WebSphere Application Server for z/OS V8.5.5, Fix Pack 4 or later (program number 5655-W65) is required for z/OS connector on WebSphere Application Server for z/OS.
- IBM 64-bit SDK for z/OS, Java Technology Edition V7.1 (program number 5655-W44) is required for z/OS connector on WebSphere Application Server for z/OS and Dynamic Workload Console reporting feature.
- For scheduling end-to-end in the distributed environment, see the *IBM Workload Scheduler Release Notes* at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045180>.

Related software

About this task

These IBM licensed programs can be used with IBM Workload Scheduler for z/OS to provide comprehensive, integrated DP operations:

- IBM Tivoli[®] NetView for z/OS (program number 5697-NV6) V6.1, or later.
- Report Management and Distribution System (RMDS) (program number 5648-048) V2.3, or later.
- Tivoli Decision Support for z/OS (program number 5698-B06) V1.8, or later.
- System Automation for z/OS Version 3 Release 3 (program number 5698-SA3) V3.3, or later.
- IBM Tivoli Monitoring V6.2.
- IBM Tivoli Output Manager V3.0, or later.

Parts and their relationships

The tracker and controller, their relationship, and how you can configure them.

About this task

IBM Workload Scheduler for z/OS consists of a base product, the *agent*, and a number of features. You need the base product to track your workload. Hereafter, you see the agent referred to as the *tracker* and the engine referred to as the *controller*. One z/OS system in your complex is designated the *controlling* system and runs the *controller* feature. Only one controller feature is required, even when you want to start standby controllers on other z/OS systems in a sysplex.

You can also control the workload in other operating environments (UNIX, Windows, IBM i) using the end-to-end scheduling functions provided in IBM Workload Scheduler for z/OS and in IBM Workload Scheduler.

Additionally, national language features let you see the IBM Workload Scheduler for z/OS ISPF dialogs in the language of your choice. These languages are available:

- English
- German
- Japanese
- Korean
- Spanish

The rest of this section describes the tracker and controller, their relationship, and how you can configure them.

Tracker

About this task

A tracker is required for every z/OS system in an IBM Workload Scheduler for z/OS configuration. The tracker handles the submission of jobs and tasks on the system, and keeps track of the status of the workload. In conjunction with standard interfaces to JES and SMF, IBM Workload Scheduler for z/OS records the relevant information about the workload by generating *event records*. The event records are captured and stored by the tracker. The tracker then communicates event information to the controller for further processing. The log where events are written by the tracker is called the *event data set*.

IBM Workload Scheduler for z/OS address spaces are defined as z/OS subsystems. The routines that run during subsystem initialization establish services that enable event information to be generated and stored in common storage (ECSA) even when an address space is not active.

You can optionally install a Data Store for each JES spool in a system. In a simple JES configuration this would mean one Data Store for each tracker. In systems with shared spools (for example, JES2 MAS), there is a Data Store for each spool, and there are fewer Data Stores than trackers.

Controller

About this task

The controller is the focal point of your IBM Workload Scheduler for z/OS configuration. It contains the controlling functions, ISPF dialogs, databases, and plans. The system that the controller is started on is called the IBM Workload Scheduler for z/OS controlling system. IBM Workload Scheduler for z/OS systems that communicate with the controlling system are called controlled or tracker systems. You need to install at least one controller for your production systems. This controls the entire IBM Workload Scheduler for z/OS configuration, the OPCplex, both local and remote.

You can use the controller to provide a single, consistent, control point for submitting and tracking the workload on any operating environment. IBM Workload Scheduler for z/OS provides distributed agents and open interfaces you use to integrate the planning, scheduling, and control of work units such as online

transactions, file transfers, or batch processing in any operating environment that can communicate with z/OS.

Server

About this task

IBM Workload Scheduler for z/OS provides a server you use to access the controller remotely from ISPF dialogs, PIFs, and the Dynamic Workload Console interface. Connections with the server run through Advanced Program-to-Program Communications (APPC) sessions or Transmission Control Protocol/Internet Protocol (TCP/IP). The server runs in its own address space; however, it is optional if you do not access the controller remotely.

The server is also used to communicate with the distributed agents for the end-to-end scheduling with fault tolerance capabilities.

Using a Started Task JCL you start and stop one or more servers either individually, using the Start and Stop operator commands, or automatically with the controller, using a keyword in the OPCOPTS statement. A server must start on the same z/OS image as its controller. Only one server can be started with the end-to-end scheduling with fault tolerance capabilities active.

The PIF dialog connection to the controller, whether via server or subsystem interface, is only allowed when the code is at the same level on both sides of the interface.

Graphical user interfaces

About this task

One graphical user interface is packaged with the product. You can use it in addition to, or in place of, ISPF:

Dynamic Workload Console

The Web-based user interface for the entire IBM Workload Automation suite of products. It is the strategic user interface for the suite and includes support for the latest functions and enhancements featured in IBM Workload Scheduler for z/OS.

The console gets access to the controller by way of the IBM Workload Scheduler for z/OS connector component, which is installed by default with the Dynamic Workload Console and is connected to IBM Workload Scheduler for z/OS by TCP/IP. For detailed information about how you install the z/OS connector with the Dynamic Workload Console, see Chapter 13, "Installing," on page 259.

The Dynamic Workload Console and IBM Workload Scheduler for z/OS connector are components that you install and run on distributed platforms, Windows, UNIX, and Linux.

One z/OS connector can be used to communicate with multiple IBM Workload Scheduler for z/OS controllers, and can serve multiple machines running Dynamic Workload Console . The graphical user interface and z/OS connector are installed in the installation directory of the same computer.

The Dynamic Workload Console manual is *Dynamic Workload Console User's Guide*.

The Dynamic Workload Console documentation is also part of the IBM Workload Scheduler library and is in the following IBM Workload Scheduler guides:

Tivoli Workload Automation: Administration Guide

Documents configuration and miscellaneous administrative tasks for Dynamic Workload Console.

Tivoli Workload Automation: Troubleshooting Guide

Documents logging and tracing Dynamic Workload Console and explains how to troubleshoot Dynamic Workload Console problems.

Tivoli Workload Automation: Messages and Codes

Documents Dynamic Workload Console messages.

These manuals are available in the IBM Workload Scheduler information center:
<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.itws.doc/toc.xml>

Additional publications specific to Dynamic Workload Console are:

Dynamic Workload Console Download Document

Provides you with detailed information about downloading the product installation images. You can access it at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039471>.

Dynamic Workload Console Detailed System Requirements

A dynamically maintained document which provides detailed information about the supported platforms, the hardware and software prerequisites, and the supported client browsers. You can access it at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045182>.

Dynamic Workload Console Release Notes[®]

A dynamically maintained document which contains the following topics:

- What is new in the release
- Interoperability tables
- Software limitations and workarounds
- Installation limitations and workarounds
- Internationalization Notes
- Documentation updates
- APARS fixed in the release

You can access it at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045183>.

You can access these documents also from the Welcome page of the IBM Workload Scheduler information center on the Web.

The Dynamic Workload Console includes also exhaustive online help as well as user scenarios and viewlets.

Data Store

About this task

Data Store is a separate address space. Its function is to collect structured (steps and data sets) and, optionally, unstructured (SYSOUT) information for all submitted jobs.

Data Store is required if you want to use the Restart and Cleanup functions:

- Restart at the job or step level
- Data set clean up
- JOBLOG retrieval

The controller can be connected to Data Store using XCF or SNA, or TCP/IP.

Output collector

About this task

The function of this started task is to collect the logs of jobs and dynamic jobs run on IBM Workload Scheduler for z/OS Agents (z-centric) and to send them to the JES spool, where they can be taken and archived by any integrated output management product into a dedicated repository.

Output collector is sent an event (in the form of a record in an event data set) by the controller every time a job or a dynamic job completes or terminates in the z-centric environment. The event contains the information necessary for the output collector to identify the job and the agent that ran it. The output collector then retrieves the job log from the agent (or the dynamic domain manager if the job is dynamic) and copies it to a SYSOUT in JES to make it available to an output management product.

In addition, Output collector attaches a header at the top of the job log with information that classifies the output (occurrence name, occurrence IA, job name, workstation name, operation number, start time, end time) and information about the run (process ID, return code, duration, status, hostname).

Activation of this feature is optional. If you activate it, it automatically collects the logs of all jobs run in the z-centric environment, regardless of whether they complete successfully or terminate in error. If you do not activate it, you can still configure your system to either request logs manually or to receive those of jobs ended in error.

In a sysplex configuration the Output collector started task must reside in the same image where the controller is.

The Output collector is fully described in *IBM Workload Scheduler for z/OS: Scheduling End-to-end with z-centric Capabilities*.

Configurations

About this task

You can configure IBM Workload Scheduler for z/OS to control virtually any combination of operating environments. IBM Workload Scheduler for z/OS can automatically schedule, submit, and track batch jobs, started tasks, and write-to-operator (WTO) messages. You can also use it to coordinate manual activities in your production workload.

Your configuration can include:

- A controlling system
- Controlled systems:
 - Local and remote controlled z/OS systems, including a parallel sysplex
 - Standby controller systems

- Previous IBM Workload Scheduler for z/OS releases
- Controlled systems running on distributed agents
- Other operating environments that do not support the distributed agents.

Chapter 2, “Planning your configuration,” on page 17 explains connections between IBM Workload Scheduler for z/OS systems and shows examples of configurations.

Controlling system

About this task

A controlling system requires both a tracker and a controller. If you install only one system, this is the controlling system. The controlling system can communicate with controlled z/OS systems using shared DASD, the cross-system coupling facility (XCF), network communication function (NCF), and Transmission Control Protocol/Internet Protocol (TCP/IP).

Controlled systems

About this task

A controlled z/OS system requires a tracker. Communication with the controlling system is through shared DASD, XCF, NCF, or TCP/IP. The tracker writes event records to an event data set, and transfers the records to the controlling system if connected using XCF, NCF or TCP/IP. NCF uses ACF/VTAM to link IBM Workload Scheduler for z/OS systems.

If you use XCF for communication, you can include a *standby controller* on one or more controlled systems. A standby controller is started in its own address space. It can take over the functions of the controller if z/OS fails or if the controller itself fails. It cannot perform the functions of a tracker while in standby mode.

The controller also controls the workload in distributed environments:

- In the end-to-end scheduling with z-centric capabilities network.
- In the end-to-end scheduling with fault tolerance capabilities network, through the end-to-end server.

IBM Workload Scheduler for z/OS: Customization and Tuning describes in detail how to control other operating environments.

Integration with IBM Workload Scheduler

About this task

Integration with IBM Workload Scheduler is provided by activating either of the following features:

End-to-end scheduling with z-centric capabilities

This feature is designed to let you schedule and control workload from the mainframe to distributed systems through z-centric agents, in a very simple architecture of the end-to-end scheduling framework. IBM Workload Scheduler for z/OS becomes the single point of control, providing you with all the mainframe capabilities to manage distributed workload. Communication between the z-centric agents and IBM Workload Scheduler for z/OS controller is direct, through the HTTP or HTTPS protocol. This feature enables dynamic scheduling of jobs (by connecting a

dynamic domain manager to the controller) as well as scheduling of job types with advanced options (file transfer, database, web services, J2EE, and more).

For detailed information about the end-to-end scheduling with z-centric capabilities, see the *IBM Workload Scheduler for z/OS: Scheduling End-to-end with z-centric Capabilities* manual.

End-to-end scheduling with fault tolerance capabilities

This feature is based on the Common Agent Technology and it enables IBM Workload Scheduler for z/OS to be the master of an IBM Workload Scheduler distributed network. This configuration is implemented by connecting an IBM Workload Scheduler domain manager directly to IBM Workload Scheduler for z/OS.

IBM Workload Scheduler for z/OS receives events from the IBM Workload Scheduler distributed network and updates the current plan (CP) according to these events. Conversely, every time the current plan is updated, an event is sent to the distributed network to update local plans on the distributed agents.

Being fault-tolerant, the distributed agents can independently continue scheduling when communications with IBM Workload Scheduler are interrupted due to network problems. At the same time, the distributed agents are prevented from acting on IBM Workload Scheduler for z/OS jobs because these are viewed as running on the Master, the only node authorized to operate on those jobs.

A CPU type named fault-tolerant workstation logically defines on IBM Workload Scheduler for z/OS each IBM Workload Scheduler agent that will be running jobs for IBM Workload Scheduler for z/OS. For detailed information about end-to-end scheduling with fault tolerance capabilities, see the *IBM Workload Scheduler for z/OS: Scheduling End-to-end with Fault Tolerance Capabilities* manual.

Subtasks

An IBM Workload Scheduler for z/OS address space (subsystem) consists of many z/OS subtasks. Some of these subtasks are always attached when the subsystem is started, others are conditionally attached according to initialization parameters specified for the scheduler options (OPCOPTS) statement in the IBM Workload Scheduler for z/OS parameter library.

Table 1 describes the subtasks.

Table 1. IBM Workload Scheduler for z/OS subtasks

Subtask ID	Component code	Description	Component of FMID	Activated by OPCOPTS parameter	Function
APPC	PP	APPC functions	JWSZ932	APPCTASK(YES)	Starts APPC support
AR	AR	Automatic recovery	JWSZ932	RECOVERY(YES)	Manages failing operations
CPH	CPH	Critical path handler	JWSZ932	Always activated	Updates the critical job table

Table 1. IBM Workload Scheduler for z/OS subtasks (continued)

Subtask ID	Component code	Description	Component of FMID	Activated by OPCOPTS parameter	Function
DRT	DX	Data router	HWSZ930	Always activated	Routes data to other subtasks or IBM Workload Scheduler for z/OS subsystems
EMGR	EM	Event manager	JWSZ932	OPCHOST(YES)	Processes job-tracking events
ERDR	ER	Event reader	HWSZ930	ERDRTASK(<i>n</i>)	Reads events from an event data set
EWTR	EW	Event writer	HWSZ930	EWTRTASK(YES)	Writes events to an event data set
EXA	EX	External router	JWSZ932	OPCHOST(YES)	Calls EQQUX009 to route submit requests to a user-defined destination ID
FL	FL	Fetch joblog	JWSZ932	RCLEANUP(YES)	Retrieves JOBLOG information
GEN	GS	General service	JWSZ932	OPCHOST(YES)	Processes IBM Workload Scheduler for z/OS dialog requests
HTC	HTC	HTTP client	JWSZ932	HTTP keyword of ROUTOPTS	Manages communications with z-centric agents through the HTTP or HTTPS protocol
HTS	HTS	HTTP client	JWSZ932	HTTP keyword of ROUTOPTS	Listens for inbound requests from the z-centric agent
ID	ID	TCP/IP Data Store	HWSZ930	TCPDEST keyword of FLOPTS	Manages communications with TCP/IP-connected Data Stores
IP	IP	TCP/IP tracker	HWSZ930	TCPIP keyword of ROUTOPTS	Manages communications with TCP/IP-connected standard trackers
JCC	JC	Job completion checker	HWSZ930	JCCTASK(YES)	Scans SYSOUT data sets
JLA	JL	JT and DB logs archiver	JWSZ932	OPCHOST(YES)	Copies JT and DB logs to the archive data set (EQQJTARC and EQQDBARC, respectively)
NMM	NM	Normal mode manager	JWSZ932	OPCHOST(YES)	Maintains the current plan
PSU	PS	Pre-SUBMIT tailoring	JWSZ932	RCLEANUP(YES)	Tailors the JCL before submitting it by adding the EQQCLEAR pre-step
RODM	RM	RODM support	HWSZ930	RODMTASK (YES)	Starts RODM support
SUB	SU	Submit task	HWSZ930	Always activated	Initiates work (job submit, job release, and WTO and STC operations)
TWS	TWS	End-to-end with fault tolerance capabilities task	JWSZ932	TPLGYSRV keyword of OPCOPTS	Handles events to and from fault-tolerant workstations (using the IBM Workload Scheduler for z/OS server)

Table 1. IBM Workload Scheduler for z/OS subtasks (continued)

Subtask ID	Component code	Description	Component of FMID	Activated by OPCOPTS parameter	Function
VTAM®	CB	Network communication function (NCF)	HWSZ930	NCFTASK(YES)	Transmits and receives IBM Workload Scheduler for z/OS data through a VTAM link
WSA	WA	Workstation analyzer	JWSZ932	OPCHOST(YES)	Schedules work for processing

Note: The subtask ID is the same identifier used to control the subtask using the z/OS MODIFY command.

When a controller is started in standby mode, only the IBM Workload Scheduler for z/OS main task (EQQMAJOR) is started. The subtasks that comprise an active controller are attached when a takeover is performed.

Relationship between the Scheduler and z/OS

Relationship between the Scheduler and z/OS.

About this task

IBM Workload Scheduler for z/OS is a z/OS subsystem, initialized during IPL. Routines run during subsystem initialization establish basic services, such as an event queue in ECSA. IBM Workload Scheduler for z/OS uses standard interfaces to SMF and JES to gather relevant information about the workload on the z/OS system.

The functions of the controller are available when an address space has been created for it, and the required subtasks have been successfully initialized. The controller can run either as a started task or as a batch address space. Normally, the address space is started during the IPL process, that is by a z/OS start command in `COMMNDm`, or by console automation. Alternatively, a z/OS operator can issue a `START` command from the operator console. The z/OS operator can also stop or modify the address space, using the `STOP` and `MODIFY` commands.

A TSO user accesses IBM Workload Scheduler for z/OS services using the dialogs. A dialog is a sequence of ISPF panels. Many of the functions supported by the dialogs pass service requests from the TSO user's address space to the controller address space for processing.

Before performing any function you request, the dialog function passes the request to the system authorization facility (SAF) router. If RACF, or a functionally equivalent security product, is installed and active on the z/OS system, the SAF router passes the verification request to RACF to perform this authority check.

A typical dialog service request is to access one or more records in VSAM files that are maintained and controlled by IBM Workload Scheduler for z/OS. Such a request is passed to IBM Workload Scheduler for z/OS through the z/OS subsystem interface (SSI). This interface invokes a routine that resides in common storage. This routine must be invoked in APF-authorized mode.

Consider that all long term plan (LTP) and CP batch planning jobs have to be excluded from SMARTBATCH DA (Data Accelerator) processing. When the SMARTBATCH DATA ACCELERATOR is used with the scheduler LTP and CP

batch planning jobs, the normal I/O to EQQCKPT is delayed until END OF JOB (or at least END OF JOBSTEP). This interferes with the normal exchange of data between the batch job and the controller started task so that when the batch job signals the controller to check the EQQCKPT to determine whether a new current plan has been created, the required updates to the CKPT have not yet been made. This causes the controller to conclude that no NCP has been created, and no turnover processing is done. As a result, even if the plan jobs run successfully, the NCP is not taken into production by the controller unless a CURRPLAN(NEW) restart is performed.

The Data Store uses the MVS/JES SAPI functions to access sysout data sets, allowing concurrent access to multiple records from a single address space.

Batch optimizer utilities, such as BMC Batch Optimizer Data Optimizer and Mainview Batch Optimizer, prevent correct communication between the scheduler's controller and CP/LTP batch planning jobs. The scheduler's logic depends on an exchange of enqueues and real-time updates of several sequential data sets to pass information back and forth between the controller's STC and the CP/LTP batch planning jobs. These optimizers hold I/O from the batch jobs until END OF STEP or END OF JOB, then preventing the required communication from taking place. When such utilities are allowed to "manage" I/O for the scheduler's CP or LTP batch planning jobs, communication between the jobs and the controller is disrupted. This causes numerous problems that are hard to diagnose. Most commonly, the CURRENT PLAN EXTEND or REPLAN jobs will run to normal completion, and an NCP data set will be successfully created, but the controller will fail to automatically take the new plan into production until it is forced to do so via a CURRPLAN(NEW) restart of the CONTROLLER. Use of BATCHPIPES with these batch planning jobs will result in the same sorts of problems.

Using the IBM Workload Scheduler for z/OS Program Directory

Using the IBM Workload Scheduler for z/OS Program Directory.

About this task

The *IBM Workload Scheduler for z/OS: Program Directory* provided with the product distribution tape might include technical information that is more recent than the information provided in this publication. In addition, the *Program Directory* describes the program temporary fix (PTF) level of the IBM Workload Scheduler for z/OS licensed program that you receive.

The *Program Directory* contains instructions for unloading the product and information about additional maintenance for your level of the distribution tape.

Before you start installing the product, check the *preventive service planning* (PSP) bucket for recommendations added by the service organizations after your *Program Directory* was produced. The PSP includes a Service Recommendations section that includes high impact or pervasive (HIPER) APARs. Ensure the corresponding PTFs are installed before you start an IBM Workload Scheduler for z/OS subsystem.

Sample library

About this task

SEQQSAMP is a library included on the distribution tape containing samples of exits, application programs, and the job control language (JCL). You can use the

samples for specific installation tasks. Appendix A, “Sample library (SEQQSAMP),” on page 355 describes the members of the SEQQSAMP library. Familiarize yourself with the contents of the SEQQSAMP library before you begin installation.

The installation process

To understand the flow of the installation, migration, and customization processes, read through this guide before you install IBM Workload Scheduler for z/OS.

The following table shows the various stages in the installation process.

Table 2. Stages summarizing the IBM Workload Scheduler for z/OS installation process

Stage	Description	For more information ...
1	Plan your configuration. You might create a diagram of your own IBM Workload Scheduler for z/OS configuration to refer to during the installation process.	Chapter 2, “Planning your configuration,” on page 17 gives examples of common configurations.
2	Plan your product installation.	Chapter 3, “Planning your installation,” on page 41 describes considerations for installing IBM Workload Scheduler for z/OS and provides a checklist for the installation tasks.
3	Install the product.	Chapter 4, “Installing,” on page 53 describes the installation tasks in detail.
4	Verify your installation.	Chapter 5, “Verifying your installation,” on page 161 describes how you can verify that IBM Workload Scheduler for z/OS is correctly installed.

When you have installed the product, you might want to include more functions. *IBM Workload Scheduler for z/OS: Customization and Tuning* explains how you do this.

Chapter 2. Planning your configuration

What to consider when planning the configuration for your installation.

This chapter describes several areas to consider when planning the configuration for your installation. It explains connections between IBM Workload Scheduler for z/OS systems and provides some examples of basic configurations. See *Customization and Tuning* for details on how to configure end-to-end scheduling in a distributed environment.

Planning considerations

IBM Workload Scheduler for z/OS must recognize when events occur; for example, when a started task or job begins to run or terminates, or when a data set has been printed. It uses JES and SMF exits to obtain this information from z/OS and to create *event records* describing the changes in the system.

The event records are stored in a sequential file called the *event data set* identified by the EQQEVDS DD name.

IBM Workload Scheduler for z/OS also uses the event data set to write checkpoint information for submission requests. The first record of the data set is used for this purpose, so the EQQEVDS DD name must be specified for all IBM Workload Scheduler for z/OS address spaces. The same data set can be used for both submit checkpointing and the event-writer subtask.

Trackers

A tracker must be installed on every z/OS system that you want IBM Workload Scheduler for z/OS to control. The tracker on each system writes events to the event data set. A subtask of the tracker, called the *event writer* performs this function. For the current plan to be updated, the event information must be communicated to, and processed by, the controller. The events are routed to the controller through the connection linking the tracker and the controller, either by an *event reader* subtask, or by requesting the event writer to queue the events immediately to the data router subtask, when the connected type is not shared DASD.

Initialization statements

IBM Workload Scheduler for z/OS initialization statements specified in the parameter library describe, among other things, the configuration of your installation.

In a shared DASD environment, an event reader subtask started at the controller reads the events from the event data set. The events are then used to update the current plan. A sequence number, specified on the ERSEQNO of the ERDROPTS initialization statement, identifies each event reader subtask. This number is used to build a DD name in the JCL procedure of the address space where the event reader is started. This DD name identifies the event data set that the event reader should process. It has the format EQQEVD nn , where nn is the sequence number of the event reader that services this event data set.

When a tracker has a non-DASD connection with the controller (that is, an XCF, NCF, or TCP/IP connection) or the tracker and controller are running in the same address space, the event writer can be used to forward events directly to the controller.

When an event writer is started with the EWSEQNO keyword on the EWTROPTS initialization statement, the event writer logs event information on the event data set and adds the event concurrently to the data router queue. The event is *not* read back from the event data set each time as it is by an event reader subtask. In this configuration, events are only read back from DASD if they need to be resent to the controller during restart processing, for example when the communication link to the controller becomes active after an outage.

For more information about the ERDROPTS and EWTROPTS initialization statements, see *IBM Workload Scheduler for z/OS: Customization and Tuning*. For detailed information about allocating event data sets, see also “Event data sets (EQQEVDS, EQQEVDM, and EQQHTTP0)” on page 113.

Communication

The data router subtask is responsible for communicating the event to the controller event manager subtask, either by XCF, NCF, TCP/IP or by adding directly to the queue when the tracker and controller are started in the same address space. This eliminates the need for a separate event-reader function, saves time, and saves I/O operations.

The EWSEQNO value is not used to build a DD name, as happens with the event reader subtask. The event writer uses the EQQEVDS DD name to identify the event data set.

If a connection is lost between a tracker and the controller, the event writer continues to record events. When the connection is restored, the event data set is processed from the last event received by the controller before the outage.

Note: Controllers scheduling work (for a given MVS™ image) must have unique subsystem names.

How to connect IBM Workload Scheduler for z/OS systems

Learn how to connect IBM Workload Scheduler for z/OS systems.

IBM Workload Scheduler for z/OS systems can be connected using any of these methods:

- Shared DASD
- XCF communication links
- VTAM link
- TCP/IP link

The controller uses any of these methods to transmit work to a tracker system. The tracker system uses the same connection to transmit events back to the controller.

Distributed agents communicate with the controller using TCP/IP services.

Shared DASD

Learn about shared DASD.

When two IBM Workload Scheduler for z/OS systems are connected through shared DASD, they share two data sets for communication:

- Event data set
- Submit/release data set

The tracker writes the event information it collects to the event data set. An event reader, started in the controller, reads the data set and adds the events to the data router queue.

A submit/release data set is one method that the controller uses to pass work to a controlled system. When two IBM Workload Scheduler for z/OS systems share a submit/release data set, the data set can contain these records:

- Release commands
- Job JCL
- Started-task JCL procedures
- Data set cleanup requests
- WTO message text

Both the host and the controlled system must have access to the submit/release data set. The EQQSUDS DD name identifies the submit/release data set in the tracker address space. At the controller, the DD name is user-defined, but it must be the same name as that specified in the DASD keyword of the ROUTOPTS statement. The controller can write to any number of submit/release data sets.

z/OS cross-system coupling facility

IBM Workload Scheduler for z/OS uses the z/OS cross-system coupling facility (XCF) to connect IBM Workload Scheduler for z/OS systems using XCF communication links.

When one or more trackers are connected to the controller through XCF communication links, the IBM Workload Scheduler for z/OS systems form an XCF group. The systems use XCF group, monitoring, and signaling services to communicate. The controller submits work and control information to the trackers using XCF signaling services. The trackers use XCF services to transmit events back to the controller.

XCF connections let IBM Workload Scheduler for z/OS support a hot standby controller and automatic-workload-restart functions.

VTAM (network communication function)

IBM Workload Scheduler for z/OS uses the network communication function (NCF) to connect a tracker to the controller using a VTAM link.

The controller transmits work to the tracker through NCF, and the same connection is used to pass back event information.

TCP/IP

IBM Workload Scheduler for z/OS uses Transmission Control Protocol/Internet Protocol (TCP/IP) to connect a tracker to the controller using a TCP/IP link.

The controller transmits work to the tracker through TCP/IP, and the same connection is used to pass back event information. The scheduler uses TCP/IP also to connect a distributed agent to the server. The TCP/IP connection between the server and the agents is established by the server.

Workstation destination

The various physical and logical locations where tasks are performed at your installation are represented in IBM Workload Scheduler for z/OS by workstations. Each workstation groups related activities. Every operation in the application description database and the current plan is associated with a workstation. You define workstations in the workstation description database.

The destination field is one attribute of a workstation description. It identifies the system in your configuration that operations scheduled for this workstation should be submitted to. The field can contain the DD name of a submit/release data set, an XCF member name, the VTAM LU of a tracker, or a user-defined destination.

If the destination field is not blank, the same name must also be present in the APPC, DASD, SNA, TCP, TCPIP, USER, or XCF keywords of the ROUTOPTS statement, depending on the connection method.

The destination field can also remain blank. A blank destination field means that operations at this workstation will be submitted by the controller or by a fault-tolerant agent, whose workstation type is FTA.

The operation-initiation exit, EQQUX009, handles the routing of the workload to user-defined destinations.

Workload restart

You can use workload restart (WLR) to restart and reroute work in your IBM Workload Scheduler for z/OS configuration. WLR tracks the status of workstations. It can be invoked when a workstation becomes inactive; that is, when the controller cannot communicate with the tracker at the destination that the workstation represents.

If an operation is *restartable*, it can be started again after a workstation failure. If an operation is *reroutable*, it can be moved to an alternative workstation for running when its workstation is no longer active.

For WLR purposes, the status of a workstation can be either active or inactive. An inactive workstation has a status of offline, failed, or unknown. The actions that WLR performs depend on the new status of the workstation and on the values you specify on the WSFAILURE and WSOFFLINE keywords of the JTOPTS initialization statement. The inactive status that a workstation can have depends on the type of connection between the tracker and the controller. The connection type and the new workstation status determine whether workload restart actions can be invoked automatically. You can use the full capabilities of WLR on systems that are connected by XCF.

Note: JES also has restart functions, which can be used when the system is restarted after a failure. JES can restart jobs that were active when the failure occurred. To prevent jobs from being started twice, ensure that both JES and WLR do not perform restart actions for jobs on the failing system.

JES considerations

The JES type and configuration in your installation has implications on your IBM Workload Scheduler for z/OS configuration.

Consider these situations in a JES type configuration:

1. On systems where JES2 is installed, an IBM Workload Scheduler for z/OS Tracker must be installed on each system in the JES2 Multi-Access Spool (MAS) complex.
2. If you do not install IBM Workload Scheduler for z/OS on all systems in a JES3 complex, ensure that:
 - A tracker is installed on the global.
 - Jobs are submitted, whether by IBM Workload Scheduler for z/OS or outside IBM Workload Scheduler for z/OS, to a system where a tracker is installed. Use the `//*MAIN SYSTEM=sysid` statement in the JCL, or start job classes used by these jobs only on those systems where a tracker is installed.
 - If you track print operations, output is printed only on those systems where a tracker is installed.

Basic server configuration example

An example of a basic server configuration of IBM Workload Scheduler for z/OS.

IBM Workload Scheduler for z/OS connects distributed agents to the server via TCP/IP. The controller transmits work to the fault-tolerant workstations via TCP/IP, and the same connection is used to pass event information back. The server is connected to the first level domain managers in the distributed network. TCP/IP is also used to connect the Dynamic Workload Console to the server through the connector. The server connects to the remote interfaces, either Programming Interfaces or remote ISPF interface users, using APPC. The following example shows a simple configuration using mixed protocols and minimal parameter customization.

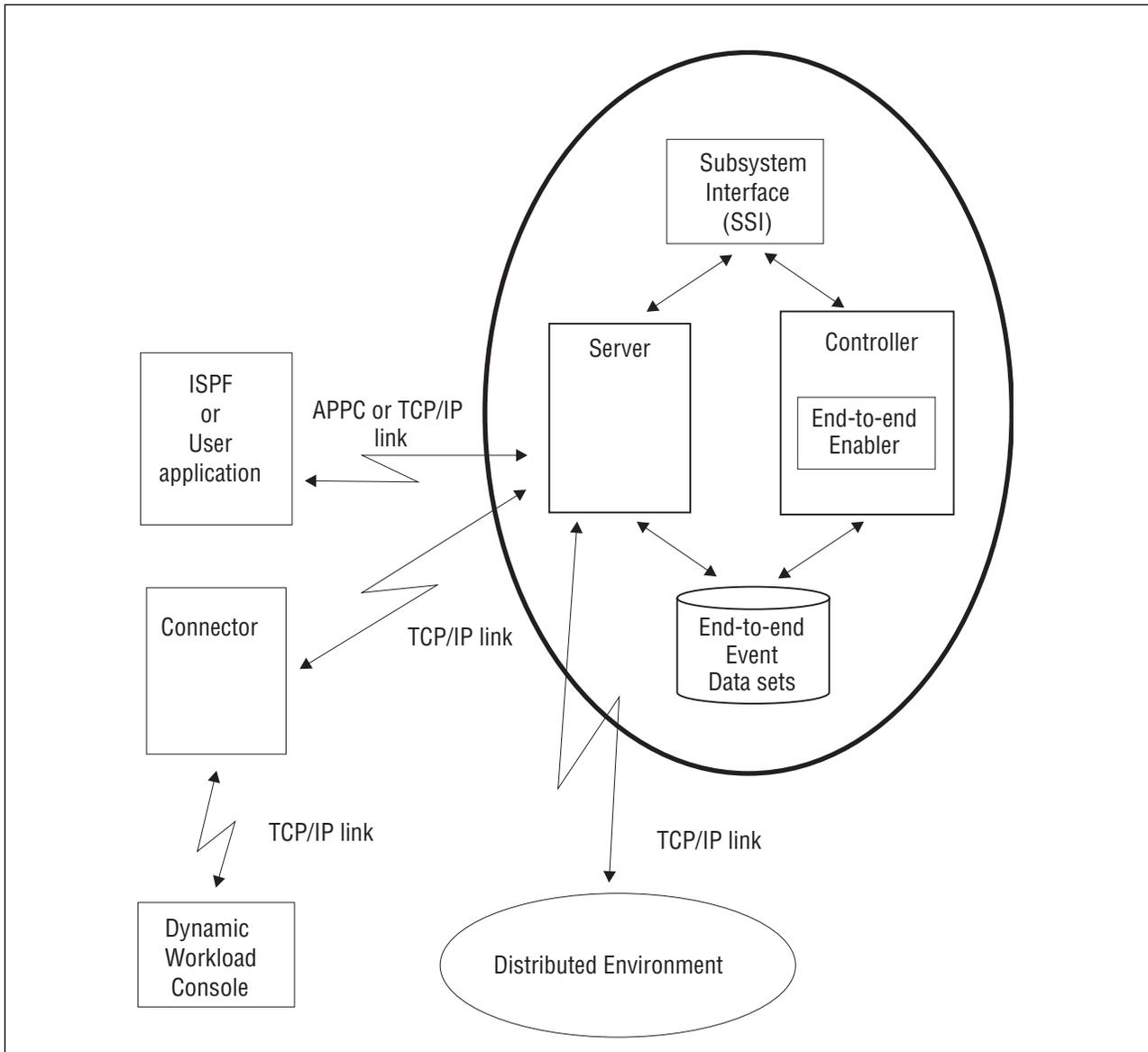


Figure 1. A basic server configuration example

Table 3 shows the initialization statements you can use to create the configuration in Figure 1, using the TCP/IP link for the user application-server communication.

Table 3. Example *EQQSERP* members for Figure 1

EQQSERP Examples	
EQQSERP	TPLGY
SERVOPTS SUBSYS(OPCA)	TOPOLOGY TPLGYMEM(TPLGYDOM)
USERMAP(USERS)	BINDIR('/usr/lpp/TWS9.3.0')
PROTOCOL(E2E,TCP)	WRKDIR('/var/TWS/OPCA')
TPLGYPRM(TPLGY)	USRMEM(TPLGYUSR)
	CODEPAGE(IBM-280)
INIT CALENDAR(DEFAULT)	

Table 3. Example *EQQSERP* members for Figure 1 (continued)

EQQSERP Examples	
TPLGYUSR	TPLGYDOM
USRREC USRCPU(FTW1) USRNAM('tws93') USRPSW('tws93')	DOMREC DOMAIN(DOM0) DOMMNGR(FTW1) DOMPARENT(MASTERDM) CPUREC CPUNAME(FTW1) CPUOS(WNT) CPUNODE('xxx.xx.xxx.x') CPUDOMAIN(DOM0) CPUTYPE(FTA) CPUTCPIP(31111) CPUFULLSTAT(ON) CPUAUTOLNK(ON) CPULIMIT(SYSTEM) FIREWALL(NO) CPUTZ('EUT')

Note: For *USERS* members, see the *SERVOPTS USERMAP* parameter, and for the *TPLGY* members, see the *TOPOLOGY* statement in *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Basic configuration examples

Examples of IBM Workload Scheduler for z/OS configurations using the various connection methods.

The examples are based on a single-image z/OS environment. Appendix B, “Configuration examples,” on page 371 contains examples of more complex configurations.

The examples in this section show:

- All IBM Workload Scheduler for z/OS address spaces as Version 2 subsystems.
- Sample initialization statements that you can use to create the configuration. Only initialization statements that specifically relate to the configuration are included.
- The IBM Workload Scheduler for z/OS components that are required, the flow of automatic work submission, and event collection in various system combinations.

DASD connected

Figure 2 on page 24 shows two IBM Workload Scheduler for z/OS address spaces with a DASD connection on a z/OS system.

You represent this system to IBM Workload Scheduler for z/OS by defining a computer workstation with a destination field that specifies a submit/release DD name. The controller writes JCL, release commands, WTO messages, and cleanup requests into the submit/release data set. The tracker reads the submit/release data set and performs the following actions:

- Submits JCL for batch jobs to the JES internal reader
- Writes the JCL for started tasks into the *EQQSTC* data set and issues *START procname z/OS* commands
- Issues JES release commands for jobs in *HOLD* status
- Submits the cleanup job.

The event-tracking routines create event records to describe activities that occur on the system. These records are added to the tracker event writer queue in ECSA. The tracker processes the queue and writes the events into the event data set. An event-reader subtask started in the controller address space reads the event data set, and the current plan is updated.

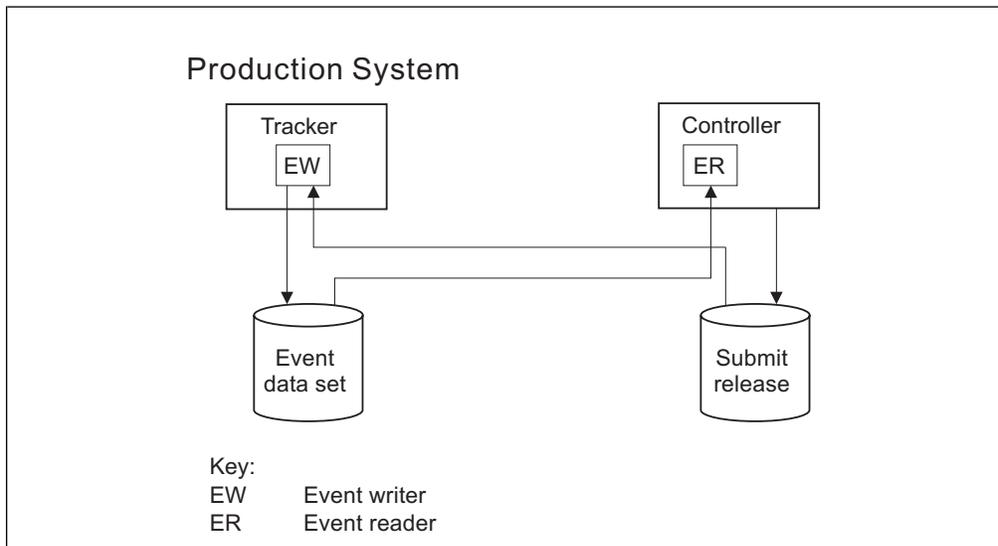


Figure 2. A z/OS system connected through shared DASD

You can also configure this system without a submit/release data set. When the workstations destination is blank; batch jobs, started tasks, release commands, and WTO messages, are processed by the submit subtask automatically started in the controller address space. The event-tracking process remains unchanged.

Table 4 shows the initialization statements you can use to create the configuration in Figure 2.

Table 4. Example EQQPARM members for Figure 2

Members for the controller	Members for the tracker
OPCECNT	TRKA
OPCOPTS OPCHOST(YES) ERDRTASK(1) ERDRPARM(STDERDR)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(STDEWTR)
ROUTOPTS DASD(EQQSYSYA)	TRROPTS HOSTCON(DASD)
STDERDR	STDEWTR
ERDROPTS ERSEQNO(01)	EWTROPTS SUREL(YES)

Note: In this example, EQQSYSYA is used for the user-defined DD name of the submit/release data set. This DD name appears in the JCL procedure of the controller and in the destination field of the workstation.

VTAM connected

Figure 3 on page 25 shows two IBM Workload Scheduler for z/OS address spaces with a VTAM connection on a z/OS system.

You represent this system to IBM Workload Scheduler for z/OS by defining a computer workstation with a destination field that specifies the LU name of the tracker. The controller transmits JCL, release commands, WTO messages, and cleanup requests across the LU-LU link using the NCF component. The tracker receives data across the VTAM link and performs the following actions:

- Submits JCL for batch jobs to the JES internal reader
- Writes the JCL for started tasks into the EQQSTC data set and issues *START procname z/OS* commands
- Issues JES release commands for jobs in HOLD status
- Submits the cleanup job.

The event-tracking routines create event records to describe activities that occur on the system. These records are added to the tracker event writer queue in ECSA. The tracker processes the queue, transmits the records to the controller across the VTAM link, and writes the events into the event data set. The VTAM subtask in the controller receives the event records, and the current plan is updated.

Note: You must specify EQQEVDS for a controller, even if an event writer is not started in the controller address space. The EQQEVDS data set is used for submit checkpointing. It can be the same data set that is used by an event-writer function. Use a unique EQQEVDS for each address space of the scheduler.

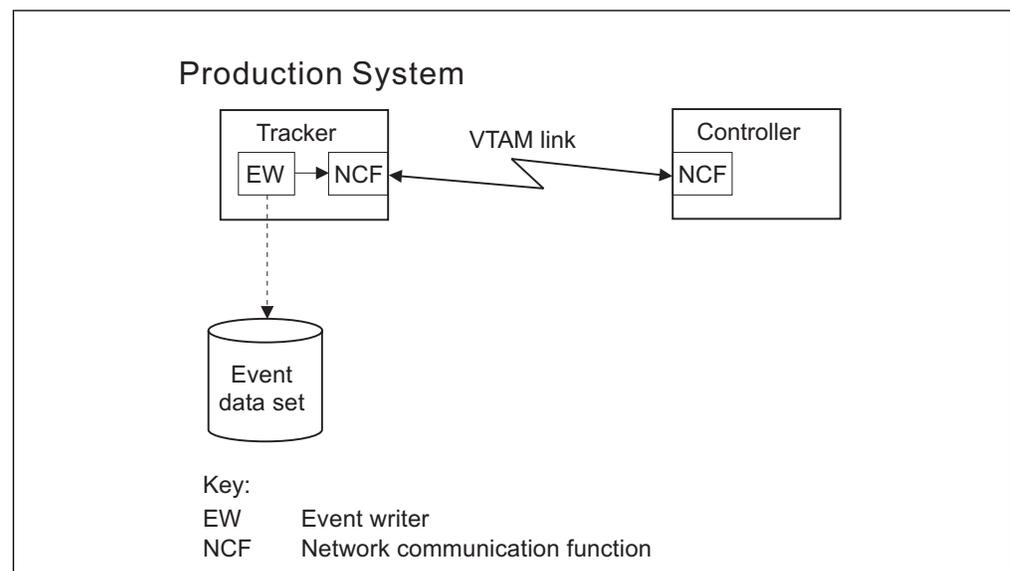


Figure 3. A z/OS system with a VTAM connection

Table 5 on page 26 shows the initialization statements you can use to create the configuration in Figure 3.

Table 5. Example EQQPARM members for Figure 3

Members for the controller	Members for the tracker
OPCECNT	TRKA
OPCOPTS OPCHOST(YES) ERDRTASK(0) NCFTASK(YES) NCFAPPL(CNTSYS)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARRM(STDEWTR)
ROUTOPTS SNA(TRKSYS)	NCFTASK(YES) NCFAPPL(TRKSYS) TRROPTS HOSTCON(SNA) SNAHOST(CNTSYS)
	STDEWTR
	EWTROPTS EWSEQNO(01)

Note: In this example, the LU name of the controller is CNTSYS and the tracker uses TRKSYS. The tracker LU is defined in the destination field of the workstation.

TCP/IP connected

Figure 4 on page 27 shows two IBM Workload Scheduler for z/OS address spaces with a TCP/IP connection on a z/OS system.

You represent this system to the scheduler by defining a computer workstation with a destination field that specifies the destination name of the tracker. The controller transmits JCL, release commands, WTO messages, and cleanup requests across the TCP/IP link. The tracker receives data across the TCP/IP link and performs the following actions:

- Submits JCL for batch jobs to the JES internal reader
- Writes the JCL for started tasks into the EQQSTC data set and issues *START procname z/OS* commands
- Issues JES release commands for jobs in HOLD status
- Submits the cleanup job.

The event-tracking routines create event records to describe activities that occur on the system. These records are added to the tracker event writer queue in ECSA. The tracker processes the queue, transmits the records to the controller across the TCP/IP link, and writes the events into the event data set. The IP task in the controller receives the event records, and the current plan is updated.

Note: You must specify EQQEVDS for a controller, even if an event writer is not started in the controller address space. The EQQEVDS data set is used for submit checkpointing. It can be the same data set that is used by an event-writer function. Use a unique EQQEVDS for each address space of the scheduler.

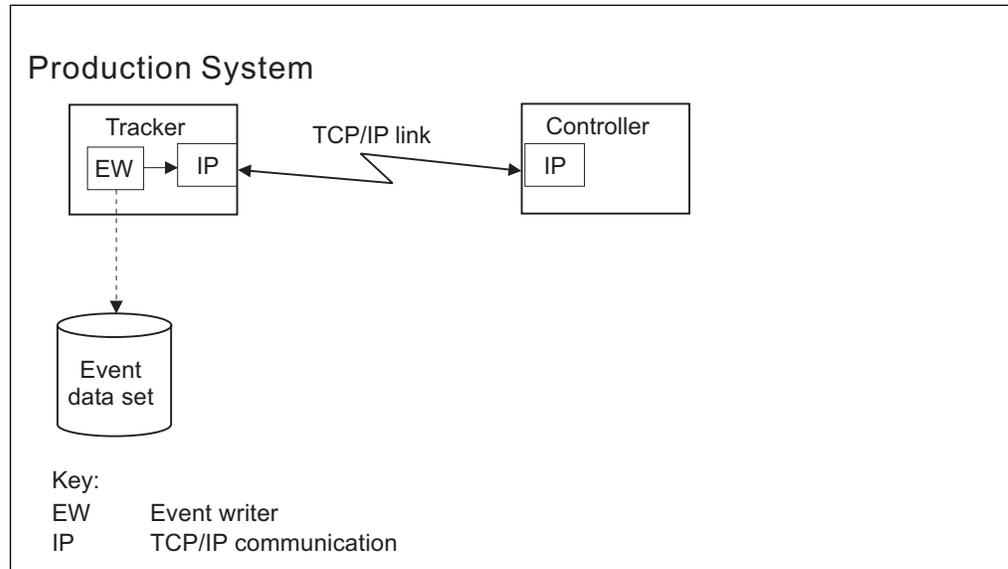


Figure 4. A z/OS system with a TCP/IP connection

Table 6 shows the initialization statements you can use to create the configuration in Figure 4.

Table 6. Example EQQPARM members for Figure 4

Members for the controller	Members for the tracker
OPCECNT	TRKA
OPCOPTS OPCHOST(YES) ERDRTASK(0)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES)
ROUTOPTS TCPIP(DEST1:'1.111.111.111'/4444)	TRROPTS HOSTCON(TCP) TCPHOSTNAME('9.12.134.1')
TCPOPTS TCPIPJOBNAME('TCPIP') HOSTNAME('9.12.134.1') TRKPORTNUMBER(8888)	TCPOPTS TCPIPJOBNAME('TCPIP') HOSTNAME('1.111.111.111') TRKPORTNUMBER(4444)
	STDEWTREWTROPTS EWSEQNO(01)

Note: In this example, the name of the destination is DEST1. The destination name is defined also in the destination field of the workstation.

XCF connected

Figure 5 on page 28 shows two IBM Workload Scheduler for z/OS address spaces with an XCF connection in a z/OS monoplex.

You represent this system to IBM Workload Scheduler for z/OS by defining a computer workstation with a destination field that specifies the XCF member name of the tracker. The controller uses XCF services to transport JCL, release commands, WTO messages, and cleanup requests to members in the sysplex. The tracker receives data from XCF and performs the following actions:

- Submits JCL for batch jobs to the JES internal reader
- Writes the JCL for started tasks into the EQQSTC data set and issues START *procname* z/OS commands
- Issues JES release commands for jobs in HOLD status

- Submits the cleanup job.

The event-tracking routines create event records to describe activities that occur on the system. These records are added to the tracker event writer queue in ECSA. The tracker processes the queue, transports the records to the controller across the XCF link, and writes the events into the event data set. The data router subtask in the controller receives the event records from XCF, and the current plan is updated.

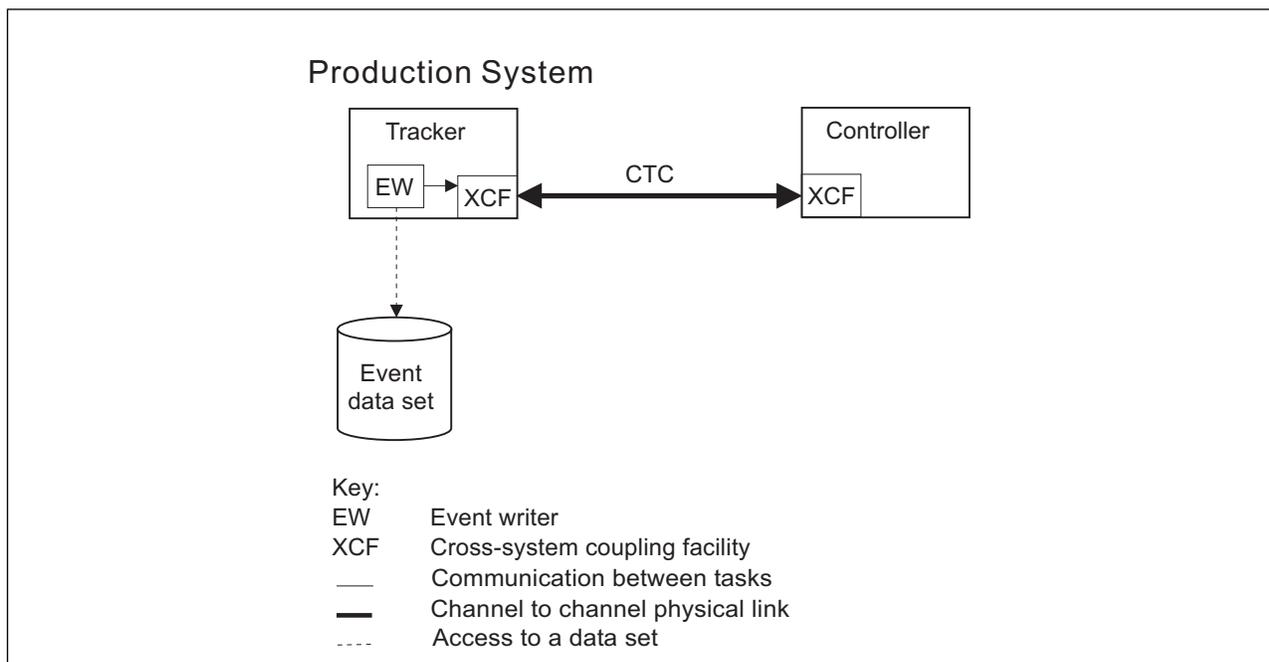


Figure 5. A z/OS system with an XCF connection

Table 7 shows the initialization statements you can use to create the configuration in Figure 5.

Table 7. Example EQQPARM members for Figure 5

Members for the controller	Members for the tracker
OPCECNT	TRKA
OPCOPTS OPCHOST(YES) ERDRTASK(0)	OPCOPTS OPCHOST(NO) ERDRTASK(0)
ROUTOPTS XCF(OPCTRK)	EWTRTASK(YES)
XCFOPTS MEMBER(OPCCNT) GROUP(PLEXSYSA)	EWTRPARM(STDEWTR)
	TRROPTS HOSTCON(XCF)
	XCFOPTS MEMBER(OPCTRK) GROUP(PLEXSYSA)
	STDEWTR
	EWTROPTS EWSEQNO(01)

Note: In this example, the name of the monoplex is PLEXSYSA. The members in that group are:

- OPCCNT**
The controller
- OPCTRK**
The tracker

The tracker member name is defined in the destination field of the workstation.

Tracker and controller in a single address space

Figure 6 shows one IBM Workload Scheduler for z/OS address space performing the function of both the tracker and the controller. To optimize availability, do not use this configuration in your production environment. However, at least one of your IBM Workload Scheduler for z/OS test environments will probably use this configuration.

You represent this system to IBM Workload Scheduler for z/OS by defining a computer workstation with a blank destination field. The submit subtask performs the following actions:

- Submits JCL for batch jobs to the JES internal reader
- Writes the JCL for started tasks into the EQQSTC data set and issues `START procname z/OS` commands
- Issues JES release commands for jobs in HOLD status

The event-tracking routines create event records to describe activities that occur on the system. These records are added to the subsystem event writer queue in ECSA. The event writer subtask processes the events and:

- Adds the event to the data router queue, and the current plan is updated
- Writes the events into the event data set.

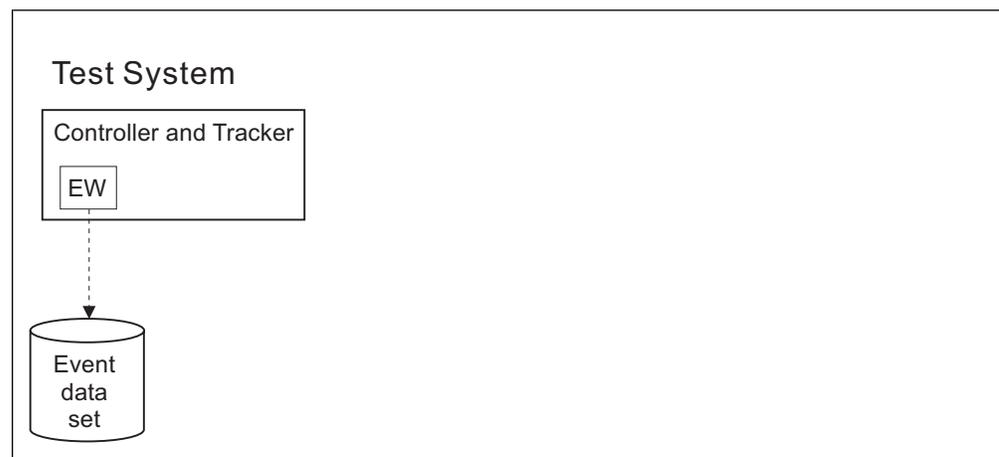


Figure 6. A tracker and controller configured in a single address space

Table 8 shows initialization statements to create the configuration in Figure 6.

Table 8. Example EQQPARM members for Figure 6

EQQPARM members for the address space	
OPCECNT	STDEWTR
OPCOPTS	OPCHOST(YES)
	ERDRTASK(0)
	EWTRTASK(YES)
	EWTRPARAM(STDEWTR)
	EWTROPTS EWSEQNO(01)

Appendix B, “Configuration examples,” on page 371 contains IBM Workload Scheduler for z/OS configuration examples for more complex environments.

Basic data store configuration examples

You need to install a Data Store for each spool tracked by IBM Workload Scheduler for z/OS in the configuration.

If you have a shared spool, for example, JES2 MAS, you can have a single Data Store for multiple trackers. Three kinds of controller-Data Store connections are supported: SNA, XCF, and TCP/IP. The Data Store type must be defined either as SNA, XCF, or TCP/IP. The same controller can connect, at the same time, with more than one Data Store, using a different connection type for each Data Store. Note that you need separate LU values: one for the Data Stores and one for the trackers. All Data Stores work on a reserved destination which must always have the same name.

SNA only connection

Figure 7 shows a JES2 with two images. In Image 1, the controller and tracker are in the same address space. Image 2 contains a tracker. The spool is not shared. Two Data Stores are required, one for Image 1 and one for Image 2. All connections are VTAM links.

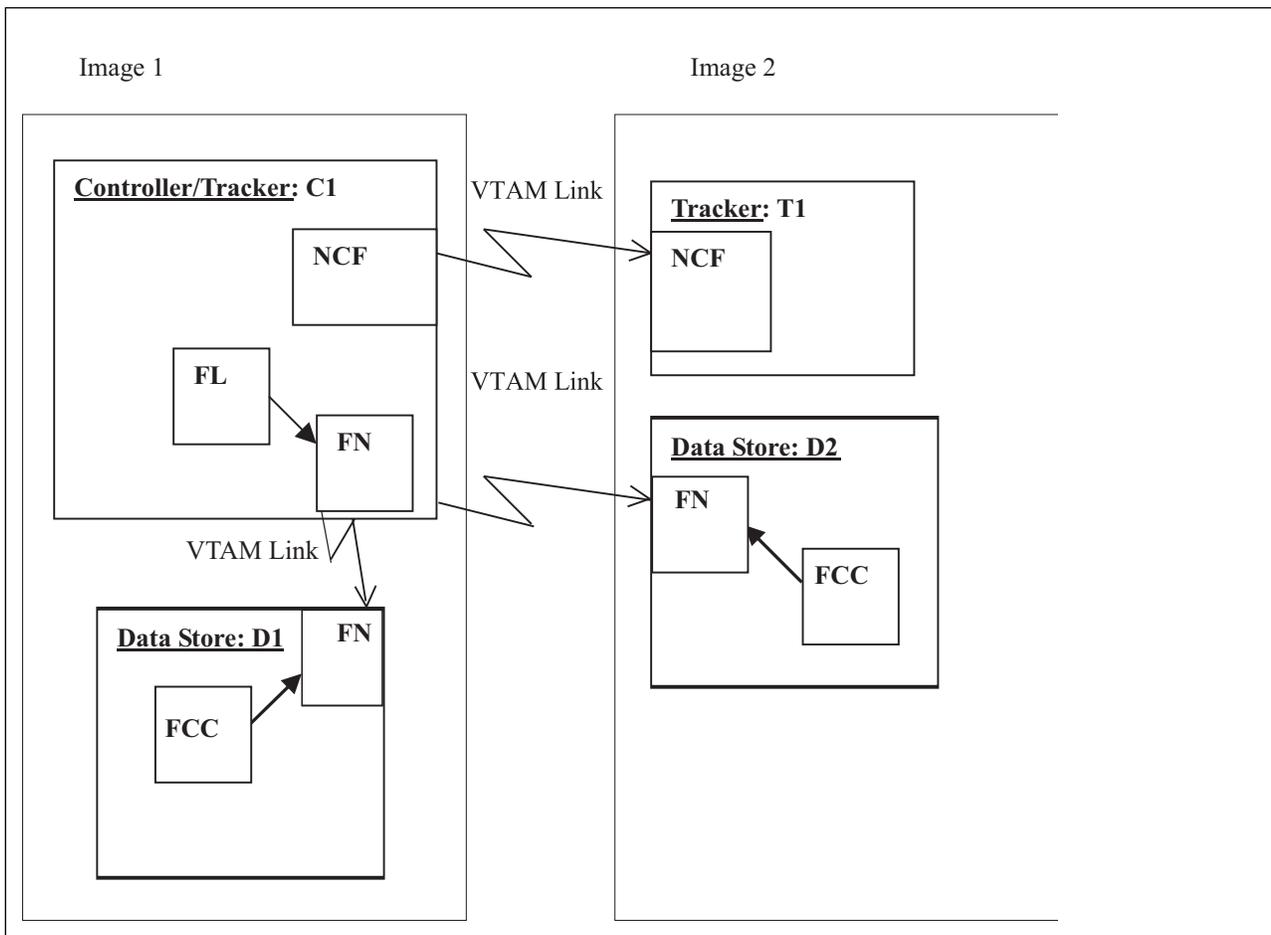


Figure 7. Controller and tracker in same address space with tracker connected through SNA

Key:

FCC Data Store Communication task

- FL** Fetch Job Log Task
- FN** Data Store SNA handler task
- NCF** Network Communication function

Table 9 shows the initialization statements you can use to create the configuration in Figure 7 on page 30.

Table 9. Example members for Figure 7

Controller member	Tracker member
C1	T1
OPCOPTS	OPCOPTS
RCLEANUP(YES)	NCFTASK(YES)
NCFTASK(YES)	NCFAPPL (LU000T1)
NCFAPPL(LU00C1T)	
	TRROPTS
FLOPTS	HOSTCON(SNA)
CTLLUNAM(LU00C1D)	SNAHOST(LU00C1T)
SNADEST(*****.LU000D1, LU000T1.LU000D2)	
ROUTOPTS SNA(LU000T1)	
D1	D2
DSTOPTS	DSTOPTS
DSTLUNAM(LU000D1)	HOSTCON(SNA)
CTLLUNAM(LU00C1D)	DSTLUNAM(LU000D2)
	CTLLUNAM(LU00C1D)
Data Store members	

Note: In this example, the LU names for the communication partners are the following:

LU00C1D

Controller C1, when communicating with a Data Store.

LU000D1

Data Store D1.

LU000D2

Data Store D2.

LU00C1T

Controller C1, when communicating with tracker T1.

LU000T1

Tracker T1.

XCF only connection

Figure 8 on page 32 shows a JES2 MAS (shared spool) with two images. In Image 1, the controller and a tracker are in the same address space and connected via XCF. Image 2 contains another tracker. You need only one Data Store, which is installed in Image 2. The controller will request the Job Log from the Data Store using the FL subtask.

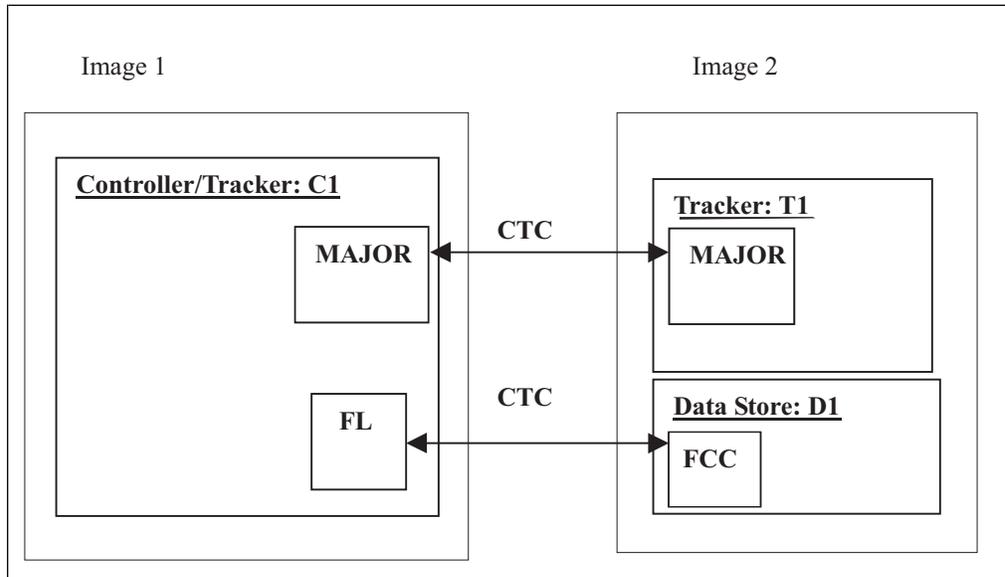


Figure 8. Controller, tracker, and Data Store connected through XCF

Key:

FL Fetch Job Log task

FCC Data Store Communication task

MAJOR
Controller/tracker main task

Table 10 shows the initialization statements you can use to create the configuration in Figure 8.

Table 10. Example members for Figure 8

Controller member	Tracker member
C1	T1
OPCOPTS	OPCOPTS
RCLEANUP(YES)	NCFTASK(NO)
NCFTASK(NO)	
ROUTOPTS	TRROPTS
XCF(XCFMEMENT1)	HOSTCON(XCF)
XCFOPTS	XCFOPTS
GROUP(XCFGRUCT)	GROUP(XCFGRUCT)
MEMBER(XCFMEMCT)	MEMBER(XCFMEMENT1)
FLOPTS	
DSTGROUP(XCFGRUCD)	
CTLMEM(XCFMEMCD)	
XCFDEST(*****.XCFMEMD1, XCFMEMENT1.XCFMEMD1)	
Data Store member	
D1	
DSTOPTS	
HOSTCON(XCF)	
DSTGROUP(XCFGRUCD)	
DSTMEM(XCFMEMD1)	

Note: In this example, the XCF groups for the communication partners are the following:

XCFGRUCD

The XCF group for the communication between controller and Data Store.
The members in the group are:

XCFMEMCD

The controller.

XCFMEMD1

The Data Store.

XCFGRUCT

The XCF group for the communication between controller and tracker. The members in the group are:

XCFMEMCT

The controller.

XCFMEMT1

The tracker.

TCP/IP only connection

Figure 9 on page 34 shows a JES2 with two images. In Image 1, the controller and tracker are in the same address space. Image 2 contains a tracker. The spool is not shared. Two Data Stores are required, one for Image 1 and one for Image 2. All connections are TCP/IP links.

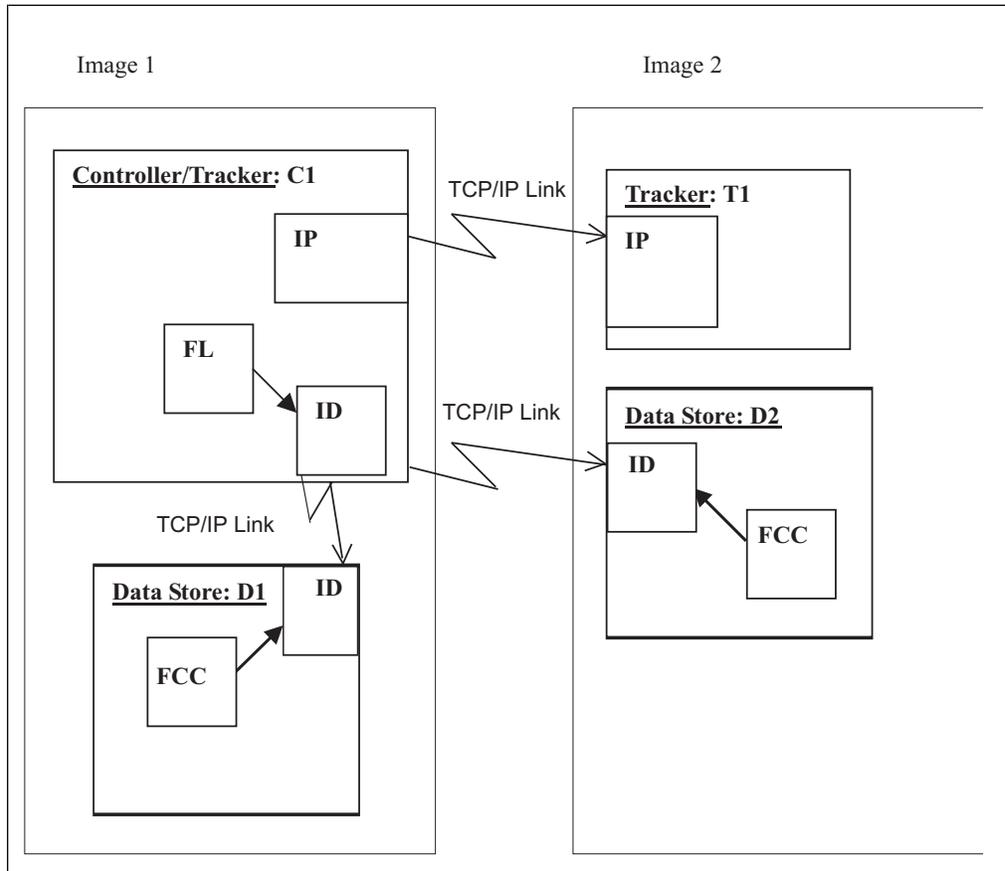


Figure 9. Controller and tracker in same address space with tracker connected through TCP/IP

Key:

FCC Data Store Communication task

FL Fetch Job Log Task

ID Task for Data Store-to-controller TCP/IP communication

IP Task for tracker-to-controller TCP/IP communication

Table 11 shows the initialization statements you can use to create the configuration in Figure 9.

Table 11. Example members for Figure 9

Controller member	Tracker member
C1	T1
OPCOPTS RCLEANUP(YES)	TRROPTS HOSTCON(TCP) TCPHOSTNAME('9.12.134.1')
FLOPTS TCPDEST(*****.'9.12.134.1', '9.12.134.9')	
ROUTOPTS TCPIP(TRK1:'9.12.134.9')	
Data Store members	

Table 11. Example members for Figure 9 (continued)

Controller member	Tracker member
D1	D2
DSTOPTS	DSTOPTS
HOSTCON(TCP)	HOSTCON(TCP)
CTLHOSTNAME('9.12.134.1')	CTLHOSTNAME('9.12.134.1')

Note: In this example, the name of the tracker destination is TRK1. The destination name is defined also in the destination field of the workstation. The TCP/IP address of image 1 is 9.12.134.1 and the TCP/IP address of image 2 is 9.12.134.9.

Mixed SNA and XCF connection

Figure 10 shows a mixed SNA and XCF connection. In Image 1, the controller and tracker are in the same address space. In Image 2, the tracker is connected using XCF. In Image 3, the remote tracker is connected using SNA with a VTAM link. The spool is only shared between Image 1 and Image 2 (JES2 MAS). You must have two Data Stores, one installed in Image 2 and one in Image 3.

Note that the controller and tracker in Image 1 must have two different LU names. For each XCF connection, there must be a different XCF group name.

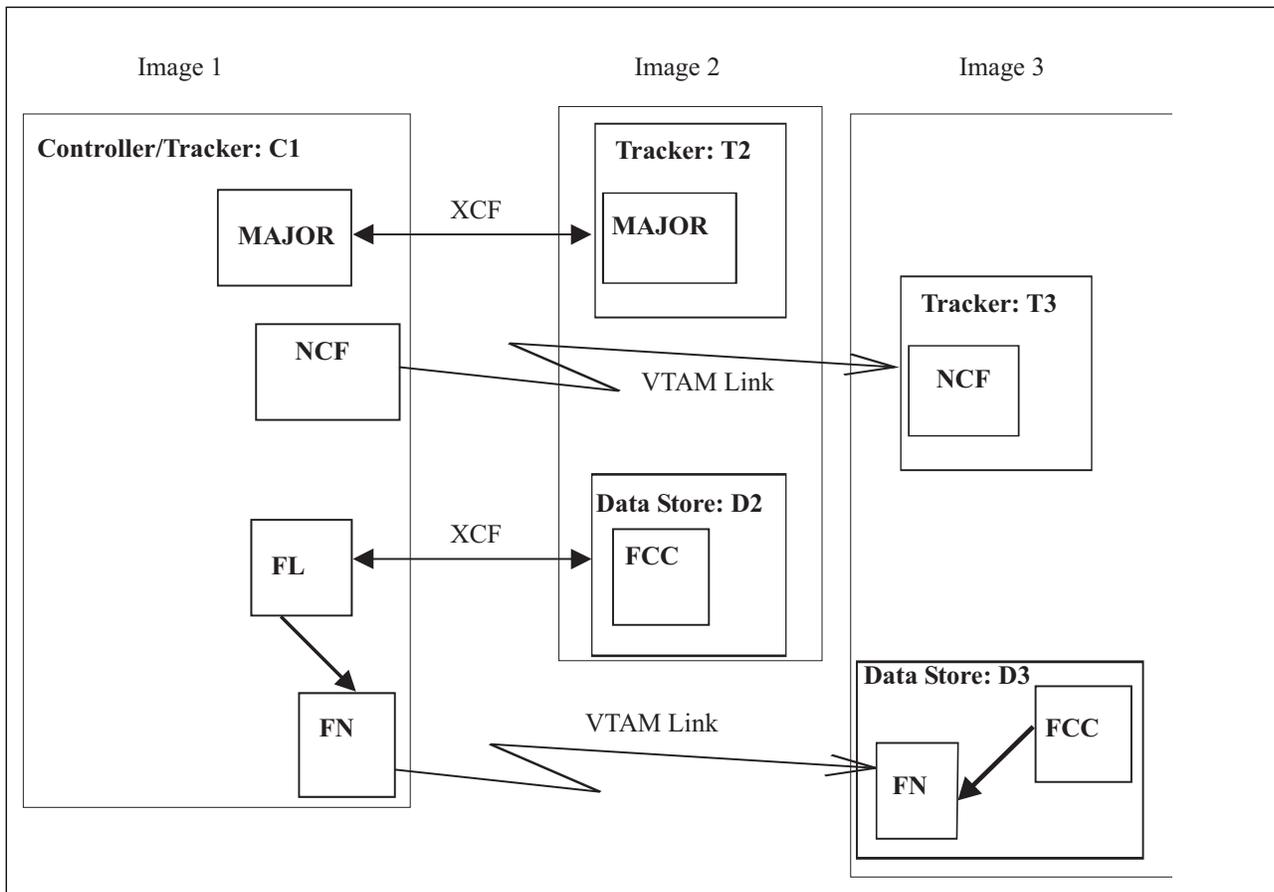


Figure 10. A mixed SNA and XCF connection

Key:

FCC Data Store Communication task

FL Fetch Job Log task

FN Data Store SNA handler task

MAJOR

Controller/tracker main task

NCF Network Communication function

Table 12 shows the initialization statements you can use to create the configuration in Figure 10 on page 35.

Table 12. Example members for Figure 10

Controller member	
C1	
OPCOPTS RCLEANUP(YES) NCFTASK(YES) NCFAPPL(LU00C1T)	
ROUTOPTS SNA(LU000T3) XCF(XCFMEMT2)	
XCFOPTS GROUP(XCFGRUCT) MEMBER(XCFMEMCT)	
FLOPTS DSTGROUP(XCFGRUCD) CTLMEM(XCFMEMCD) XCFDEST(*****.XCFMEMD2, XCFMEMT2.XCFMEMD2) CTLLUNAM(LU00C1D) SNADEST(LU000T3.LU000D3)	
Tracker members	
T2	T3
OPCOPTS NCFTASK(NO)	OPCOPTS NCFTASK(YES) NCFAPPL(LU000T3)
TRROPTS HOSTCON(XCF)	TRROPTS HOSTCON(SNA) SNAHOST(LU00C1T)
XCFOPTS GROUP(XCFGRUCT) MEMBER(XCFMEMT2)	
Data Store members	
D2	D3
DSTOPTS HOSTCON(XCF) DSTGROUP(XCFGRUCD) DSTMEM(XCFMEMD2)	DSTOPTS DSTLUNAM(LU000D3) CTLLUNAM(LU00C1D)

Note: In this example, the XCF groups or the LU names for the communication partners are the following:

XCFGRUCD

The XCF group for the communication between controller and Data Store.
The members in the group are:

XCFMEMCD

Controller C1.

XCFMEMD2

Data Store D2.

XCFGRUCT

The XCF group for the communication between controller and tracker. The members in the group are:

XCFMEMCT

Controller C1.

XCFMEMENT2

Tracker T2.

LU00C1D

Controller C1, when communicating with D3.

LU00D3

Data Store D3.

LU00C1T

Controller C1, when communicating with T3.

LU00T3

Tracker T3.

Mixed TCP/IP and XCF connection

Figure 11 on page 38 shows a mixed TCP/IP and XCF connection. In Image 1, the controller and tracker are in the same address space. In Image 2, the tracker is connected using XCF. In Image 3, the remote tracker is connected using TCP/IP. The spool is shared only between Image 1 and Image 2 (JES2 MAS). You must have two Data Stores, one installed in Image 2 and one in Image 3.

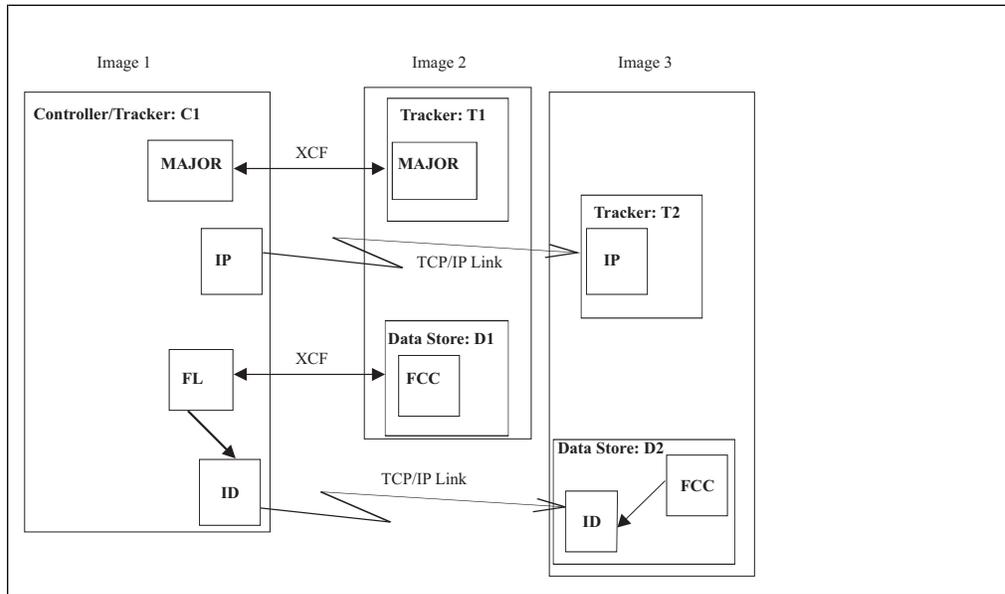


Figure 11. A mixed TCP/IP and XCF connection

Key:

FCC Data Store Communication task

FL Fetch Job Log task

FN Data Store SNA handler task

MAJOR
Controller/tracker main task

ID Task for Data Store-to-controller TCP/IP communication

IP Task for Tracker-to-controller TCP/IP communication

Table 13 shows the initialization statements you can use to create the configuration in Figure 11.

Table 13. Example members for Figure 11

Controller member
C1
OPCOPTS RCLEANUP(YES)
FLOPTS TCPDEST(*****.'1.22.333.4','1.22.333.4')
DSTGROUP(XCFGRUCD) CTLMEMT(XCFMEMCD) XCFDEST(*****.XCFMEMD2,XCFMENT2.XCFMEMD2)
ROUTOPTS TCPIP(TRK1:'1.22.333.4') XCF(XCFMEMCT)
Tracker members

Table 13. Example members for Figure 11 (continued)

Controller member	
T1	T2
OPCOPTS NCFTASK(NO)	OPCOPTS NCFTASK(NO)
TRROPTS HOSTCON(XCF)	TRROPTS HOSTCON(TCP) TCPHOSTNAME('1.22.333.4')
XCFOPTS GROUP(XCFGRUCT) MEMBER(XCFMEMT2)	
Data Store members	
D1	D2
DSTOPTS HOSTCON(XCF) DSTGROUP(XCFGRUCD) DSTMEM(XCFMEMD2)	DSTOPTS HOSTCON(TCP) CTLHOSTNAME('1.22.333.4')

Chapter 3. Planning your installation

What to consider when planning your installation on IBM Workload Scheduler for z/OS.

About this task

This chapter offers considerations for installing IBM Workload Scheduler for z/OS and provides a checklist that you can use as you work through the installation process. Chapter 4, “Installing,” on page 53 describes the installation tasks in detail.

Installation considerations

About this task

During the planning stage of your IBM Workload Scheduler for z/OS project, consider carefully how you want to install the scheduler to control your production workload. The installation consists of installing the tracker and controller in combinations to suit your processing environment, and connecting them using one or more of the communication methods described in Chapter 2, “Planning your configuration,” on page 17. Later, you can customize your IBM Workload Scheduler for z/OS systems to include more functions.

Before you start the installation tasks, ensure that you have all the resources you need to complete your installation.

Configuring for availability

About this task

It is recommended that you install the tracker and controller as separate subsystems on an IBM Workload Scheduler for z/OS controlling system. The tracker can then continue to collect events even when the controller is stopped. Events are created by SMF and JES exits, and added to a queue in the z/OS extended common service area (ECSA). If the event writer is not active while work is running in the system, the queue might fill up, and new events might be lost. IBM Workload Scheduler for z/OS cannot recover these lost events.

You can improve IBM Workload Scheduler for z/OS availability using the z/OS Automatic Restart Manager (ARM). Automatic restart management can reduce the impact of an unexpected error on IBM Workload Scheduler for z/OS because it can restart it automatically, without operator intervention.

To use Automatic Restart Manager, set the ARM parameter in the OPCOPTS statement to YES. For details about the ARM parameter, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Hot standby

About this task

If you connect your IBM Workload Scheduler for z/OS systems using the z/OS cross-system coupling facility (XCF), you can include one or more standby controllers in your configuration. A standby system can take over the functions of

the controller if the controller fails or if the z/OS system that it was active on fails. You can create a standby controller on one or more IBM Workload Scheduler for z/OS controlled systems within the XCF group. Each standby system must have access to the same resources as the controller. These resources include data sets and VTAM cross-domain resources. However, if EQQMLOG is allocated as a data set, it cannot be shared between the controller and standby controller. The standby system is started the same way as the other IBM Workload Scheduler for z/OS address spaces, but is not activated unless a failure occurs or unless it is directed to take over through a z/OS operator modify command. If you use one or more servers to remotely access the controller or to schedule using the end-to-end with fault tolerance capabilities feature, note that the server must always run on the same system as the controller.

Starting an event writer with an event reader function

About this task

In situations where a tracker does not have a DASD connection with the controller, use an event writer that is started with an event-reader function.

This can improve performance because events are not written to the event data set and then read back again, which requires an event-reader task to continually check the event data set for new events.

Instead, the event writer writes events to the event data set and forwards them directly to the controller through an XCF, NCF, or TCP/IP link.

Using a Hierarchical File System cluster

About this task

If you plan to install the end-to-end scheduling with fault tolerance capabilities feature, consider that the server starts multiple tasks and processes using the UNIX System Services (USS) on z/OS. The End-to-End server accesses USS in a Hierarchical File System cluster, that can be either HFS or zFS. For details, see Table 14 on page 44.

Using two message log (MLOG) data sets

About this task

All the major components and tasks of IBM Workload Scheduler for z/OS log messages to either SYSOUT or to a data set, based on your configuration choice.

In the case the messages are logged to a data set (EQQMLOG), the message log data set remains under the control of the IBM Workload Scheduler for z/OS controller for as long as the controller is active. If you want to save, or clear, the contents of the data set, you must first stop the controller. Also, if the controller is not stopped for a long period of time, and EQQMLOG runs out of space, the messages are written into the system log.

To avoid these problems, including the increased amount of time required to find specific records in an oversized data set, you can configure IBM Workload Scheduler for z/OS to use two message log data sets, EQQMLOG and EQQMLOG2.

The two MLOGs are used alternatively: while one logs messages, the other remains inactive. Then, when the active data set reaches a pre-set level of completion, its contents are copied into a GDG data set and it goes idle, while the other data set starts logging. When the same level of completion is reached in the now active data set, the switch is repeated. Every time the data sets are shifted, message:

```
EQQZ402I SWITCH FROM ddn1 TO ddn2
```

is issued on SYSLOG and written in the just switched MLOG.

The messages copied from EQQMLOG and EQQMLOG2 are available in the GDG data set.

To configure IBM Workload Scheduler for z/OS to use two MLOGS, specify Y in the MLOG SWITCH field of the EQQJOBSC installation aid panel. To activate this feature, you must also specify in the Switchlimit field the number of records written in the MLOG file that you want to trigger the switch mechanism. When the file reaches this level of capacity, its contents are saved in the GDG file, and the oncoming messages are written on the alternate file.

The number of records in Switchlimit must be greater than 0 for the feature to be activated.

Specifying Y in MLOG SWITCH results in the creation of sample JCLs that:

- Add DD EQQMLOG2 in the controller started task JCL (samples EQQCONO and EQQCON) and initial parameters (samples EQQCONOP and EQQCONP)
- Add DD EQQMLOG2 in the tracker started task JCL and initial parameters (samples EQQTRA and EQQTRAP)
- Allocate the EQQMLOG2 data set like EQQMLOG (sample EQQPCS02)
- Create the GDG data set where the outgoing MLOG file is archived (samples EQQSMLOG and EQQREPRO)
- Allocate the GDG root to archive the MLOGS (sample EQQPCS12)

You must then make the EQQSMLOG procedure (copied from the EQQJOBS customized samples) accessible to the controller, for example by adding it in the user.proclib.

After the product is installed, you can configure the MLOGPROCNAME and SWITCHMLOGLIM keywords of the OPCOPTS initialization statement to set or change the feature (for detailed information about OPCOPTS, see *Customization and Tuning*).

Particular attention should be paid to calibrating the number of records specified in Switchlimit (or SWITCHMLOGLIM), the point being that an exceedingly low value triggers the switching function with unnecessary frequency while at the other end you risk filling the MLOG to its capacity and generate a B37 abend. As a best practice, you enter a number that at least doubles the number of records logged at the controller startup or during the workload peak (when many ETT add messages are sent).

For example, if you use a 3390 disk, a track is 56664 bytes. One cylinder includes 15 tracks, which are 849960 bytes. If you allocate an MLOG of 1 cylinder with 1 extent, the number of available bytes is 1699920. Since the MLOG record length is 125 bytes, the maximum number of available records is 13599 (1699920/125). Under these conditions, the appropriate number of records for Switchlimit or SWITCHMLOGLIM should be a little lower, perhaps 13000. If you then find that a

greater number of messages is issued at short intervals during peak workload or at controller startup, this implies that you should increase the value and allocate more space for EQQMLOG.

Another advantage of using the MLOG switching function is that it prevents the loss of the messages that are written to EQQMLOG when abend B37 occurs as the MLOG file becomes full. In fact with the switching function on, if the current MLOG file fills up due to an inadequate number of records defined in SwitchLimit or SWITCHMLOGLIM, when B37 is issued, a switch MLOG action is forced, all records are copied to the GDG data set, and control is passed to the alternate MLOG.

When this feature is installed, you can:

- Run the modify command, /F *subsys*, SWITCHMLOG, to force the switch to the alternate data set (EQQMLOG or EQQMLOG1), regardless of the number of currently logged messages, and starts the record counter from 0 again.
- See which of the two MLOG data sets is in current use on the BROWSING GENERAL CURRENT PLAN INFORMATION dialog (EQQSGCPP; fast path 6.6).

See also “Message log data set (EQQMLOG)” on page 119.

Checklist for installing IBM Workload Scheduler for z/OS

This section contains a checklist to guide you through the installation tasks for a tracker, a controller, a standby controller, or the ISPF dialogs.

Note: Always install the tracker first on the controlling system or on a system where a standby controller will be installed.

In the checklist, the task numbers are arranged in a recommended order but are not meant to imply a required order. You perform the tasks suited to your own configuration.

The **Applies to** column indicates for which IBM Workload Scheduler for z/OS address space you should perform that particular task. You might not need to perform every task outlined in the list. Skip those tasks or actions that do not apply to your installation.

A check mark (✓) in the **IPL** column means that an IPL of the z/OS system is needed for the change to take effect. It does not indicate how many IPLs are needed. You can install IBM Workload Scheduler for z/OS with only one IPL of the z/OS system by performing all the required steps before a scheduled IPL.

The **TopicPage** column indicates the topicpage in this guide where the task is described.

Table 14. Checklist for installing IBM Workload Scheduler for z/OS

Task	Description	Applies to	IPL	Page
1	<p><i>Load tracker software.</i></p> <p>Perform these actions on each system in your IBM Workload Scheduler for z/OS configuration:</p> <ul style="list-style-type: none"> • Run SMP/E to receive tracker software. • Apply tracker maintenance. 	tracker		“Step 1. Loading tracker software” on page 55

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
2	<p>Load controller software.</p> <p>Perform these actions on each system where you are installing a controller, standby controller, or dialogs:</p> <ul style="list-style-type: none"> • Run SMP/E to receive controller software. • Apply controller maintenance. 	<p>Controller Standby controller Dialogs</p>		<p>“Step 2. Loading controller software” on page 55</p>
3	<p>Load national language support (NLS) software for the controller.</p> <p>Perform these actions on each system where you are installing a controller standby controller, or dialogs:</p> <ul style="list-style-type: none"> • Run SMP/E to receive NLS software. • Apply NLS maintenance. 	<p>Controller Standby controller Dialogs</p>		<p>“Step 3. Loading national language support software” on page 56</p>
4	<p>Run the EQQJOBS installation aid.</p> <p>You can run EQQJOBS as soon as the tracker software is loaded. It helps you install IBM Workload Scheduler for z/OS:</p> <ul style="list-style-type: none"> • Set up EQQJOBS. • Create the sample job JCL. Do this to generate tailored samples from the EQQJOBS dialog. • Generate batch job skeletons. Use EQQJOBS to generate skeletons for the ISPF dialogs. • Optionally generate the Data Store samples if you want to install the Data Store. 	<p>Tracker controller Standby controller Dialogs</p>		<p>“Setting up the EQQJOBS installation aid” on page 57</p>
5	<p>Add SMF and JES event tracking exits.</p> <p>Perform this task on every z/OS system in your IBM Workload Scheduler for z/OS configuration.</p> <p>Note: If you place exits in a link-pack-area (LPA) library, you must perform an IPL of the z/OS system with the CLPA option.</p>	<p>tracker</p>	<p>✓</p>	<p>“Step 5. Adding SMF and JES exits for event tracking” on page 81</p>

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
6	<p>Update SYS1.PARMLIB.</p> <p>On each system where you are installing the product, perform the actions that are applicable to your installation:</p> <ul style="list-style-type: none"> • Define IBM Workload Scheduler for z/OS subsystems (IEFSSN<i>nn</i>). This is required for each system where the product is installed. • Authorize the IBM Workload Scheduler for z/OS load module library (IEAAPF<i>nn</i>). Do this if you install the product in a separate load module library. • Update SMF parameters (SMFPRM<i>nn</i>). Do this when installing a tracker. • Update dump-content definitions. Consider this on each system where you are installing the product. • Update the z/OS link-library definition (LNKLST<i>nn</i>) on each system where you are installing the product. • Update XCF initialization options (COUPLE<i>nn</i>). Review this section if you use XCF connections. • Modify TSO parameters (IKJTSON<i>nn</i>). Do this when installing a controller, a standby controller, or the ISPF dialogs. • Update PPT for performance (SCHED<i>nn</i>) on each system where you are installing the product. • Define the DLF exit for Hiperbatch support (COFDLF<i>nn</i>). Do this if you use Hiperbatch support. • Choose whether to start IBM Workload Scheduler for z/OS automatically (COMMND<i>nn</i>). Consider this on each system where you are installing the product. • Update APPC options (APPCPM<i>nn</i>). Consider this action if you intend to use the IBM Workload Scheduler for z/OS API or server. Define VTAM resources before you update SYS1.PARMLIB. Coordinate this action with task 18 or 19. 	<p>Tracker controller Standby controller Dialogs</p>	✓	<p>“Step 6. Updating SYS1.PARMLIB” on page 83</p>
7	<p>Set up the RACF environment.</p> <p>Perform these actions on each system in your IBM Workload Scheduler for z/OS configuration:</p> <ul style="list-style-type: none"> • Update RACF for IBM Workload Scheduler for z/OS started tasks (ICHRIN03) on all IBM Workload Scheduler for z/OS started tasks on each system. • Update RACF for a controller or standby controller. 	<p>Tracker controller Standby controller Dialogs</p>	✓	<p>“Step 7. Setting up the RACF environment” on page 93</p>
<p>At this point, if you placed exit modules in LPA, you can IPL with CLPA. No other options for IBM Workload Scheduler for z/OS require an IPL.</p>				

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
8	<p>Set up the SSL environment</p> <p>Perform these actions to activate a secure communication in a TCP/IP network:</p> <ul style="list-style-type: none"> • Create the SSL work directory. • Create as many private keys, certificates, and trusted certification authority (CA) chains as you plan to use in your network. • Configure the scheduler, by specifying the TCPOPTS statement for each component of your network. 	<p>Tracker controller Standby controller Data Store server User address space</p>		<p>“Step 8. Securing communications” on page 99</p>
9	<p>Allocate data sets.</p> <p>Perform these actions if you are installing a tracker or a controller:</p> <ul style="list-style-type: none"> • Review the section on allocating IBM Workload Scheduler for z/OS data sets. Do this before you allocate data sets. • Allocate VSAM data sets for a controller. Perform this action to create new data sets for a controller. • Allocate non-VSAM data sets. Perform this action for each IBM Workload Scheduler for z/OS address space. • Optionally allocate the VSAM Data Store data sets if you want to use the Data Store. • Optionally allocate the files and directory to use the end-to-end scheduling with fault tolerance capabilities. 	<p>Tracker controller Data Store server</p>		<p>“Step 9. Allocating data sets” on page 102</p>
10	<p>Update SYS1.PROCLIB.</p> <p>Perform these actions for each IBM Workload Scheduler for z/OS address space.</p> <ul style="list-style-type: none"> • Create a JCL procedure for each address space on all z/OS systems where you are installing IBM Workload Scheduler for z/OS. • If you use IBM Workload Scheduler for z/OS to schedule started-task operations, ensure that the started-task-submit data set (EQQSTC) is in the JES PROCLIB concatenation and in the master scheduler start procedure. • If you use Restart and Cleanup, copy the EQQCLEAN sample procedure to a data set that is referenced in the JES PROCLIB concatenation. 	<p>Tracker controller Standby controller</p>		<p>“Implementing support for started-task operations” on page 129</p>
11	<p>Define initialization statements.</p> <p>Perform this task for each IBM Workload Scheduler for z/OS address space:</p> <ul style="list-style-type: none"> • Define initialization statements. Create members in the parameter library (EQQPARM) for each address space. 	<p>Tracker controller Standby controller</p>		<p>“Step 11. Defining the initialization statements” on page 134</p>

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
12	<p>Create a DB2 database.</p> <p>Perform this task if you need history support:</p> <ul style="list-style-type: none"> • Update initialization statements. • Create a DB2 database. 	Controller Standby controller		“Step 12. Creating the DB2 database” on page 134
If you are not using NCF, XCF, or TCP/IP connections you can now start a tracker and continue with the verification task.				
13	<p>Set up the ISPF environment.</p> <p>Perform these actions on the system where you are installing the ISPF dialogs.</p> <ul style="list-style-type: none"> • Set up the IBM Workload Scheduler for z/OS CLIST library. • Set up the ISPF tables. • Allocate ISPF and IBM Workload Scheduler for z/OS data sets to your TSO session. • Invoke the IBM Workload Scheduler for z/OS dialog. 	Dialogs		“The ISPF environment” on page 188
If you are not using NCF, XCF, or TCP/IP connections, the API or server, you can now start a controller and continue with the verification task.				
14	<p>Using XCF for local communications.</p> <p>Even if you have already specified XCF initialization statements in Task 12 and updated the COUPLEnn member in Task 7, consider these actions if you use XCF:</p> <ul style="list-style-type: none"> • Update XCF initialization options. Ensure that XCF initialization options are suitable for your IBM Workload Scheduler for z/OS configuration. • Add initialization statement options for XCF. Specify XCF runtime options in the parameter library for all started tasks. 	Tracker controller Standby controller		“Step 14. Using XCF for communication” on page 141
15	<p>Activate NCF for VTAM connections.</p> <p>Perform these actions for each address space that is connected through NCF. Ensure that a standby controller has the same tracker connections as the controller and that each tracker can connect to all standby controllers:</p> <ul style="list-style-type: none"> • Add NCF network definitions. Define VTAM applications on each system where a started task uses NCF. • Add NCF session parameters on each z/OS system where IBM Workload Scheduler for z/OS is installed. • Update the COS table. Consider this action if you do not want to use the VTAM COS default entry. • Activate network resources. Check that all the required VTAM resources are active. • Add NCF initialization options. Include initialization statement options in the parameter library for all started tasks that use NCF. 	Tracker controller Standby controller		“Step 15. Activating the network communication function” on page 143

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
16	<p>Activate TCP/IP connections</p> <p>Perform these actions for each address space that is connected via TCP/IP. Ensure that a standby controller has the same tracker connections as the controller and that each tracker can connect to all standby controllers:</p> <ul style="list-style-type: none"> • Add TCP/IP network definitions. Define IP address for the controller and tracker. • Add TCP/IP initialization options. Include initialization statement options in the parameter library for all started tasks that use TCP/IP. • For TCP/IP, the IBM Workload Scheduler for z/OS server can manage up to 500 concurrent connection requests in a queue. In the PROFILE.TCPIP configuration file, set the SOMAXCONN statement to a value not greater than 500. 	<p>Tracker controller Standby controller</p>		<p>“Step 16. Using TCP/IP for communication” on page 146</p>
17	<p>Activate support for the IBM Workload Scheduler for z/OS API.</p> <p>To use the API, perform these actions for each IBM Workload Scheduler for z/OS address space that you want to send requests to:</p> <ul style="list-style-type: none"> • Define VTAM resources. Define a local LU for IBM Workload Scheduler for z/OS, logon modes, and cross-domain resources, as required. • Update APPC options. Update the APPCPMnn member of SYS1.PARMLIB. • Activate IBM Workload Scheduler for z/OS support for APPC. In the parameter library, include APPCTASK(YES) on the OPCOPTS statement. 	<p>Tracker controller Standby controller</p>		<p>“Step 17. Activating support for the API” on page 147</p>
18	<p>Activate support for the IBM Workload Scheduler for z/OS Server to use APPC or TCP/IP communications.</p> <p>Note: Include this task when activating an IBM Workload Scheduler for z/OS server.</p> <p>To activate the server, perform these actions in the order shown:</p> <ol style="list-style-type: none"> 1. Define VTAM resources. Define a local LU for IBM Workload Scheduler for z/OS, logon modes, and cross-domain resources, as required. 2. Update APPC options. Update the APPCPMnn member of SYS1.PARMLIB. 3. Activate IBM Workload Scheduler for z/OS support for APPC. In the parameter library, include SERVERS on the OPCOPTS statement. 	<p>Server controller Standby controller</p>		<p>“Step 18. Activating support for the product dialog and programming interface using the server” on page 150</p>

Table 14. Checklist for installing IBM Workload Scheduler for z/OS (continued)

Task	Description	Applies to	IPL	Page
19	<p>Activate support for end-to-end scheduling with fault tolerance capabilities</p> <p>Note: Include this task when you intend to use IBM Workload Scheduler for z/OS to schedule jobs on distributed fault-tolerant agents.</p> <ul style="list-style-type: none"> • Ensure that you have loaded the fault-tolerant end-to-end enabler software on the system where you have installed the controller. • Verify that all the VSAM and non-VSAM data sets and the files used for the end-to-end scheduling with fault tolerance capabilities have been allocated (for details, see the task that describes how to allocate data sets). • To activate the server, include TPLGYPRM on the SERVOPTS statement in the IBM Workload Scheduler for z/OS parameter library. • To activate the controller, include TPLGYSRV on the OPCOPTS statement in the IBM Workload Scheduler for z/OS parameter library. • To activate the Daily Planning batch jobs, include TPLGYPRM in the BATCHOPTS statement in the IBM Workload Scheduler for z/OS parameter library. 	Controller Standby controller server		“Step 19. Activating support for the end-to-end scheduling with fault tolerance capabilities” on page 154
20	<p>Activate support for end-to-end scheduling with z-centric capabilities</p> <p>Note: Include this task when you intend to use IBM Workload Scheduler for z/OS to schedule jobs on distributed z-centric agents.</p> <ul style="list-style-type: none"> • Define the ROUTOPTS initialization parameters for the controller. • Define the HTTPOPTS initialization parameters for the tracker. 	Tracker controller Standby controller		“Step 20. Activating support for end-to-end scheduling with z-centric capabilities” on page 155
21	<p>Activate Support for the Dynamic Workload Console</p> <p>Note: Include this task when activating a server and intending to use the Dynamic Workload Console.</p> <p>To activate the server, perform these actions:</p> <ul style="list-style-type: none"> • Install the Connector • In the parameter library, include SERVERS on the OPCOPTS statement. 	Server controller Standby controller		“Step 21. Activating support for Dynamic Workload Console” on page 155
22	<p>Activate Support for the Java™ utilities</p> <p>Perform this task if you want to use one of the following features:</p> <ul style="list-style-type: none"> • Dynamic Workload Console reporting. • Event-driven workload automation for data set triggering, with centralized deploy process. 	Controller		“Step 22. Activating support for the Java utilities” on page 157

Part 2. Installing IBM Workload Scheduler for z/OS

This part describes how to create or upgrade IBM Workload Scheduler for z/OS.

Chapter 4. Installing

How to install the current version of IBM Workload Scheduler for z/OS.

This chapter describes the tasks you perform to install IBM Workload Scheduler for z/OS. Before you begin these tasks, you must determine the IBM Workload Scheduler for z/OS configuration you want to create and the functions you want to use. See Chapter 2, “Planning your configuration,” on page 17. Table 15 summarizes the installation tasks and identifies the address space type for each task. Depending on your configuration, you might not need to perform every task indicated in the table. Skip those sections that are not relevant to your installation.

Note: To generate the skeletons that are appropriate to the current version, you are required to run EQQJOBS. For details, see “Step 4. Using the EQQJOBS installation aid” on page 56.

Table 15. IBM Workload Scheduler for z/OS installation tasks

Installation task	Perform for					
	tracker	Controller	Server	Standby controller	Data Store	Dialogs
“Step 1. Loading tracker software” on page 55	✓	✓		✓	✓	✓
“Step 2. Loading controller software” on page 55		✓		✓		✓
“Step 3. Loading national language support software” on page 56		✓		✓		✓
“Step 4. Using the EQQJOBS installation aid” on page 56		✓			✓	✓
“Step 5. Adding SMF and JES exits for event tracking” on page 81	✓					
“Step 6. Updating SYS1.PARMLIB” on page 83	✓	✓		✓		✓
“Step 7. Setting up the RACF environment” on page 93	✓	✓		✓		
“Step 8. Securing communications” on page 99	✓	✓	✓	✓	✓	✓
“Step 9. Allocating data sets” on page 102				✓	✓	
• “Allocating the VSAM data sets” on page 102		✓				
• “Allocating non-VSAM data sets” on page 109	✓	✓				
• “Allocating the files and directories” on page 124			✓			
“Step 10. Creating JCL procedures for address spaces” on page 128	✓	✓		✓	✓	
“Step 11. Defining the initialization statements” on page 134	✓	✓		✓	✓	

Table 15. IBM Workload Scheduler for z/OS installation tasks (continued)

Installation task	Perform for					
	tracker	Controller	Server	Standby controller	Data Store	Dialogs
"Step 12. Creating the DB2 database" on page 134		✓		✓		
"The ISPF environment" on page 188						✓
"Step 14. Using XCF for communication" on page 141	✓	✓		✓	✓	
"Step 15. Activating the network communication function" on page 143	✓	✓		✓	✓	
"Step 16. Using TCP/IP for communication" on page 146	✓	✓		✓		
"Step 17. Activating support for the API" on page 147		✓		✓		
"Step 18. Activating support for the product dialog and programming interface using the server" on page 150		✓		✓		
"Step 19. Activating support for the end-to-end scheduling with fault tolerance capabilities" on page 154		✓	✓	✓		
"Step 20. Activating support for end-to-end scheduling with z-centric capabilities" on page 155	✓	✓		✓		
"Step 21. Activating support for Dynamic Workload Console" on page 155		✓		✓		
"Step 22. Activating support for the Java utilities" on page 157		✓				
Chapter 5, "Verifying your installation," on page 161						
• "Verifying installation of a tracker" on page 161	✓					
• "Verifying installation of a controller and dialogs" on page 167		✓				
• "Verifying installation of a standby controller" on page 170				✓		
• "Verifying installation of the Restart and Cleanup function" on page 172		✓		✓	✓	
• "Verifying configuration" on page 174	✓	✓		✓		
• "Step 23. Activating support for FIPS standard over SSL secured connections" on page 158		✓				

Step 1. Loading tracker software

You must load tracker software on each z/OS system in your configuration. Process the software distribution tape using the facilities of System Modification Program Extended (SMP/E). This creates or updates the necessary software libraries on your system. Table 16 describes the distribution and target libraries that are created or updated by SMP/E.

Table 16. tracker libraries loaded by SMP/E

SMP/E DD name		Description
Distribution	Target	
AEQQPNL0	SEQQPNL0	Panels for the EQQJOBS installation aid
AEQQMOD0 (object)	SEQQLMD0 (load)	tracker modules
AEQQMSG0	SEQQMSG0	Messages
AEQQMAC0	SEQQMAC0	Assembler macros
AEQQCLIB	SEQQCLIB	IBM Workload Scheduler for z/OS CLISTs
AEQQSAMP	SEQQSAMP	Sample exits, programs, and JCL
AEQQSKL0	SEQQSKL0	JCL skeletons, input to EQQJOBS
AEQQTBL0	SEQQTBL0	ISPF tables
AEQQDATA	SEQQDATA	Sample IBM Workload Scheduler for z/OS databases
AEQQMISC	SEQQMISC	DBRM files

It is recommended that you place all the IBM Workload Scheduler for z/OS load modules in a separate library. Use the same library for both the tracker and the controller. Create the library before you run the SMP/E jobs.

Alternatively, you can place the IBM Workload Scheduler for z/OS load modules in one of your existing load-module libraries, for example SYS1.LINKLIB. The remaining data sets loaded by SMP/E are new data sets that you must create before running the SMP/E jobs. The *IBM Workload Scheduler for z/OS: Program Directory* contains the JCL and instructions for loading the software.

When you have loaded the tracker software, apply any recommended maintenance described in the PSP bucket.

Step 2. Loading controller software

To load controller software, process the software distribution tape using SMP/E. This creates or updates the necessary disk-resident libraries on your system. Table 17 describes the data set that is created or updated by SMP/E.

Table 17. Controller libraries loaded by SMP/E

SMP/E DD name		Description
Distribution	Target	
AEQQMOD0 (object)	SEQQLMD0 (load)	Controller modules

Table 17. Controller libraries loaded by SMP/E (continued)

SMP/E DD name		Description
Distribution	Target	
AEQQMISC	SEQQMISC	Batch command interface tool modules and Control Language tool modules

Recommendation: Install the controller in the same library that you are using for the tracker.

When you have loaded the controller software, apply any recommended maintenance described in the PSP bucket.

Step 3. Loading national language support software

To load national language support (NLS) software, process the software distribution tape using SMP/E. This creates or updates the necessary disk-resident libraries on your system. Table 18 describes the data sets that are created or updated by SMP/E.

Table 18. NLS libraries loaded by SMP/E

SMP/E DD name		Description
Distribution	Target	
AEQQPxxx	SEQQPxxx	Panels
AEQQMxxx	SEQQMxxx	Messages
AEQQLxxx	SEQQLxxx	Advanced ISPF panel templates
AEQQGxxx	SEQQGxxx	Advanced ISPF panels
<p>Note: The suffix xxx is the NLS identifier. It is recommended that you place NLS software in the same library that you are using for the tracker and controller.</p>		

When you have loaded the national language support software, apply any recommended maintenance described in the PSP bucket.

Step 4. Using the EQQJOBS installation aid

EQQJOBS is a CLIST-driven ISPF dialog that can help you install IBM Workload Scheduler for z/OS. You can set up EQQJOBS as soon as the tracker software has been installed. EQQJOBS assists in the installation of the tracker and controller by building batch-job JCL that is tailored to your requirements. You can use this JCL to perform the following actions:

- Install the tracking exits
- Set up RACF security
- Create data sets
- Create started-task JCL
- Perform planning functions

Set up EQQJOBS now so that it is ready to use when needed.

Setting up the EQQJOBS installation aid

EQQJOBS reads skeleton JCL from the SEQQSAMP or SEQQSKL0 libraries, tailors the JCL, and then writes the tailored JCL to an output library that you specify. The output library must be a partitioned data set with record length 80 and record format FB, and must be allocated before you run EQQJOBS. The components of EQQJOBS reside in these libraries:

SEQQCLIB

CLIST to drive the dialog

SEQQPNL0

EQQJOBS panels

SEQQSAMP

Sample JCL

SEQQSKL0

IBM Workload Scheduler for z/OS batch-job skeletons.

To be able to run EQQJOBS, allocate these libraries to the DD statements in your TSO session:

- SEQQCLIB to SYSPROC
- SEQQPNL0 to ISPLIB
- SEQQSKL0 and SEQQSAMP to ISPSLIB.

To invoke EQQJOBS, enter the TSO command EQQJOBS from an ISPF environment. This panel is displayed:

```
EQQJOBS0 ----- EQQJOBS APPLICATION MENU -----
Select option ==>
                                     Userid  - SYSPROG
                                     Time     - 15:04
                                     Terminal - 3278

  1 - Create sample job JCL

  2 - Generate batch-job skeletons

  3 - Generate Data Store samples

  4 - Generate WAPL samples

  X - Exit from the EQQJOBS dialog
```

Figure 12. EQQJOBS0 - EQQJOBS application menu

The following sections describe:

- Option 1, “Creating the sample job JCL” on page 58
- Option 2, “Generating batch-job skeletons” on page 69
- Option 3, “Generating Data Store samples” on page 74
- Option 4, “Creating the Workload Automation Programming Language samples” on page 78

Note: To ensure that all files are correctly allocated, perform first option 2, followed by option 1.

Creating the sample job JCL

To ensure that all the files are correctly allocated, before creating the sample job JCL you must have generated the batch-job skeletons, as described in “Generating batch-job skeletons” on page 69.

To create the sample job JCL:

1. Select option 1 from the EQQJOBS application menu. The following panel is displayed:

```
EQQJOBS3 ----- CREATE SAMPLE JOB JCL -----
Command ==>

The data set names specified on this panel should be fully qualified
names without any enclosing apostrophes.

Enter the name of the output library:

Sample job JCL      ==> CCOPC.OPCA.INSTJCL_____

Job statement information:

==> //SYSPROG1 JOB (111111,2222),'OPCESA BATCH',CLASS=B,MSGCLASS=H,_____
==> //          MSGLEVEL=(1,1),NOTIFY=SYSPROG_____
==> //*_____
==> //*_____

The following data set names are used by one or more of the generated job

Message library name ==> OPC.SEQMSG0_____
Data library name    ==> TWSDEV.DATA_____
Parameter library    ==> CCOPC.OPCA.PARM_____
Checkpoint data set  ==> CCOPC.OPCA.CKPT_____
Press ENTER to continue
```

Figure 13. EQQJOBS3 - Create sample job JCL

The data set names that you specify on this panel must be fully-qualified and not be enclosed within apostrophes.

Sample job JCL

Required input. The name of the library to which you want that the generated JCL samples are written. The library must be allocated before you generate the batch JCL samples. Ensure that the library that you specify has sufficient directory blocks to store all the sample members that are generated by EQQJOBS (for details, see Table 19 on page 67.)

Job statement information

Required input. The JOB statement that follows standard JCL syntax and your installation standards.

Message library name

Required input. The name of the library that contains the IBM Workload Scheduler for z/OS messages (SMP/E target DDNAME SEQMSG0).

Data library name

Required input. The name of the library that will contain the SSL certificates that are provided with IBM Workload Scheduler for z/OS. For detailed information about these certificates, see the *IBM Workload Scheduler for z/OS: Scheduling End-to-end with z-centric Capabilities* manual.

Parameter library

Required input. The name of the library that will contain the initialization statements. This library will be allocated by the EQQPCS01 batch job.

Checkpoint data set

Required input. The name of the checkpoint data set. This library will be allocated by the EQQPCS01 batch job.

2. Press Enter. The following panel is displayed:

```
EQQJOBS4 ----- CREATE SAMPLE JOB JCL -----
Command ==>

Enter the following required job stream parameters:

non-VSAM dsn prefix  ==> CCOPC.OPCA _____
VSAM dsn prefix     ==> CCOPC.OPCAV _____
Unit name           ==> 3390 _____      Default unit name
Primary volume serial ==> PROD01 _____ Primary volume serial for VSAM
Backup volume serial ==> PROD02 _____ Secondary volume serial for VSAM
SYSOUT class        ==> * _____        SYSOUT class for reports

The following information is optional:

STEPLIB dsname      ==> _____
OPC.SEQQLMD0 _____
VSAMCAT dsname      ==> _____

VSAM password       ==> _____
Dsn prefix of old   ==> _____
VSAM files          ==> CCOPC.OPCAV _____
non-VSAM files      ==> CCOPC.OPCA _____
Samples with cloning support generated:  N (Y/N)

Static symbol used  ==> SYSCONE without enclosing '&' and period

Press ENTER to continue
```

Figure 14. EQQJOBS4 - Create sample job JCL

non-VSAM dsn prefix

Required input. The qualifiers that prefix the non-VSAM data set names. IBM Workload Scheduler for z/OS adds a low-level qualifier to the prefix to uniquely identify the non-VSAM data sets. For example, it adds EV for the event data set. In this example, the full name is CCOPC.OPCA.EV.

Note: IBM Workload Scheduler for z/OS does not use the prefix for the parameter library or checkpoint data set. You entered the names of these data sets, fully qualified, on the previous panel.

VSAM dsn prefix

Required input. The qualifiers that prefix the VSAM data set names. IBM Workload Scheduler for z/OS adds a low-level qualifier to the prefix to uniquely identify each IBM Workload Scheduler for z/OS VSAM data set. For example, it adds AD for the application description data set. In this example, the full name is CCOPC.OPCAV.AD.

Unit name

Required input. The device name that is valid at your installation; it can be a device type, for example 3380, or a group name, for example PROD or TEST.

Primary volume serial

Required input. The volume to be used by sample job EQQPCS01 to allocate the primary data sets. Some IBM Workload Scheduler for z/OS logical files are implemented as two physical data sets, a primary and an alternate; for example, the current plan data set. To minimize the potential impact of errors on a particular device, allocate the primary and alternate data sets on different physical devices.

SYSOUT Class

Required input. The SYSOUT class that you want to use for the reports that are generated by the sample jobs.

STEPLIB dsname

Optional. The name of the IBM Workload Scheduler for z/OS load module library if the load modules are not in a data set included in an active LNKLST member.

VSAMCAT dsname

Optional. The name of a catalog in which VSAM data sets are to be defined if they are not to be defined in the master catalog.

VSAM password

Optional. The password for the VSAM catalog if the catalog is password-protected.

VSAM files

Optional. The qualifiers that prefix your existing IBM Workload Scheduler for z/OS VSAM data set names. These are used to create the data set-conversion sample JCL.

non-VSAM files

Optional. The qualifiers that prefix your existing IBM Workload Scheduler for z/OS non-VSAM data set names. These are used to create the data set-conversion sample JCL.

Samples with cloning support generated

Optional. Enter Y if you want the SYSCONE variable resolved.

Note:

- a. Generated JCLs do not contain a period before &SYSCONE.
 - b. &SYSCONE variable is intended to be substituted in the scheduler started tasks. It is not substituted in the generated JCLs that run as batch jobs. To obtain the variable substitution, run the JCL as cataloged procedure.
3. Press Enter. The following panel is displayed:

```

EQQJOBS8 ----- CREATE SAMPLE JOB JCL -----
Command ==>

END-TO-END WITH FAULT TOLERANCE:  Y      (Y= Yes ,N= No)
Installation Directory             ==> /usr/lpp/TWS/V9R1M0_____
                                   ==> _____
                                   ==> _____
Work Directory                     ==> /var/TWS/inst_____
                                   ==> _____
                                   ==> _____
User for OPC address space         ==> UID_
Refresh CP group                   ==> GID_
RESTART AND CLEANUP (Data Store)  Y      (Y= Yes ,N= No)
Reserved destination              ==> OPCX_____
Connection type                   ==> SNA_____ (SNA/XCF/TCP)
SNA Data Store luname             ==> I9PC45AA (only for SNA connection )
SNA FN task luname               ==> I9PC45RA (only for SNA connection )
Xcf Group                         ==> _____ (only for XCF connection )
Xcf Data Store member            ==> _____ (only for XCF connection )
Xcf FL task member               ==> _____ (only for XCF connection )
TCP Data store host name:         ==> _____ (only for TCP connection )
==> _____
TCP Data store port number        ==> _____ (only for TCP connection )
Press ENTER to continue

```

Figure 15. EQQJOBS8 - Create sample job JCL

END-TO-END WITH FAULT TOLERANCE

Specify Y if you want to schedule jobs on fault-tolerant agent workstations.

Only the Server requires a UID and a GID. Set the UID to a nonzero value, unless you plan to run the EQQPCS05 sample JCL.

Installation Directory

The path where SMP/E has installed the IBM Workload Scheduler for z/OS files for UNIX system services that apply the end-to-end enabler feature. This directory is the one containing the bin directory. The default path is /usr/lpp/TWS/VvRrMm.

Work Directory

The directory where the subsystem-specific files are located. Replace /inst with a name that uniquely identifies your subsystem. Each subsystem that uses the fault-tolerant workstations must have its own work directory. Only the server and the daily planning batch jobs go in the work directory. For details, see “Allocating the files and directories” on page 124.

User for OPC address space

This information is used to create the procedure to build the directory with the right ownership. To run the end-to-end scheduling with fault tolerance capabilities correctly, the ownership of the work directory, and the files it contains, must be assigned to the same user ID that RACF associates with the Server Started Task. In the User for OPC address space field, specify the RACF user ID used for the Server address space. This is the name specified in the started-procedure table.

Refresh CP group

This information is used to create the procedure to build the directory with the right ownership. To create the new Symphony file, the user ID used to run the daily planning batch job must belong to the group that you are specifying in this field. Also ensure that the user ID associated

with the Server address space (the one specified in User for OPC address space field) belongs to this group or has this group as supplementary group.

RESTART AND CLEANUP (Data Store)

Specify Y if you want to use the Restart and Cleanup function.

Note: The panel shown in Figure 15 on page 61 creates only one sample entry, but you can define all kinds of connections, including a combination of mixed connections.

Reserved destination

The destination reserved for Data Store output. It must be the same as that for controller and Data Store parameters.

Connection type

Required input. The connection method used to handle communication between FN/FL tasks and Data Store. It can be SNA, XCF, or TCP.

SNA Data Store luname

If you chose SNA in Connection type, specify the Data Store VTAM connection name.

SNA FN task luname

If you chose SNA in Connection type, specify the VTAM application name of the controller FN task.

Xcf Group

If you chose XCF in Connection type, specify the name of the XCF group.

Xcf Data Store member

If you chose XCF in Connection type, specify the name of the Data Store XCF member.

Xcf FL task member

If you chose XCF in Connection type, specify the name of the controller FL task XCF member.

TCP Data store host name

The Data Store TCP/IP host name if you chose TCP in Connection type.

TCP Data store port number

The Data Store TCP/IP port number if you chose TCP in Connection type.

4. Press Enter. The following panel is displayed:

```

EQQJOBS9 ----- CREATE SAMPLE JOB JCL -----
Command ==>

JAVA UTILITIES ENABLEMENT:  Y      (Y= Yes ,N= No)
Installation Directory      ==> /usr/lpp/TWS/V9R1M0_____
                           ==> _____
                           ==> _____
Java Directory              ==> /usr/lpp/java/J5.6_____
                           ==> _____
                           ==> _____
Work Directory              ==> /var/TWS/inst_____
                           ==> _____
                           ==> _____
User ID                     ==> UID_____
Group ID                    ==> GID_____

JZOS Batch Launcher
PDSE Library                ==> _____
Load Module Name            ==> JVMLDM66

Press ENTER to continue

```

Figure 16. EQQJOBS9 - Create sample job JCL

JAVA UTILITIES ENABLEMENT

Specify Y to enable one or both the following features:

- Dynamic Workload Console reporting
- Event-driven workload automation feature for data set triggering

Installation Directory

The directory, with its complete path, where the product specific HFS or ZFS files are stored. This path corresponds to the path where the binary files are located, omitting the subdirectory /bin.

Java Directory

The HFS or ZFS directory where the Java Software Development Kit (SDK) for z/OS is installed.

Work Directory

The directory where the subsystem specific HFS or ZFS files are stored. Each subsystem supporting the JAVA utility must have its own work directory.

User ID

The UNIX System Services user ID.

Group ID

The UNIX System Services group ID.

JZOS Batch Launcher PDSE Library

The PDSE that contains the JZOS Batch Launcher JVMLDM module.

JZOS Batch Launcher Load Module Name

Specify JVMLDM66, that is the JZOS Batch Launcher load module name used with 64-bit SDK for z/OS 6.0.

5. Press Enter. The following panel is displayed:

```

EQQJOBSC ----- CREATE SAMPLE JOB JCL -----
Command ==>

SSL FOR TCP/IP CONNECTION      Y      (Y= Yes ,N= No)
SSL Work Directory             ==> /var/TWS/inst/ssl_____
                               ==> _____
                               ==> _____

SSL User ID                    ==> UID_____
SSL Group ID                   ==> GID_____

OUTPUT COLLECTOR               -      (Y= Yes ,N= No)
Class                          ==> _____
Writer                         ==> _____

MLOG SWITCH                    -      (Y= Yes ,N= No)
Switchlimit                    ==> _____
                               (0 means function is inactive;
                               other values represent the
                               number of records that must
                               be reached in the current MLOG
                               file to switch to the alternate
                               MLOG)

EXTENDED AUDITING              ==> -      (Y= Yes ,N= No)

STEP AWARENES                  Y      (Y= Yes ,N= No)
Store on primary controller    ==> Y      (Y= Yes ,N= No)
Press ENTER to continue

```

Figure 17. EQQJOBSC - Create sample job JCL

SSL FOR TCP/IP CONNECTION

Specify Y if you have trackers connected to the controller through TCP/IP to protect the communication with SSL.

SSL Work Directory

The directory, with its complete path, where the HFS or ZFS files containing the SSL certificates are stored. The default value is the same work directory used for the fault-tolerant end-to-end scheduling, including the /ssl subdirectory.

SSL User ID

The RACF user ID that takes the ownership of the SSL work directory. The default value is the same user ID specified for the fault-tolerant end-to-end scheduling.

SSL Group ID

The RACF group that takes the ownership of the SSL work directory. The default value is the same group specified for the fault-tolerant end-to-end scheduling.

OUTPUT COLLECTOR

If you plan to use the Output collector started task to collect job logs from IBM Workload Scheduler for z/OS Agents, specify Y to generate the related samples.

Class The name of the JES CLASS where the job logs are to be copied by Output collector. If you do not specify this value, no JESCLASS keyword is recorded in the generated EQQOUCP sample and you can define it later in the OUCOPTS customization statement (see *Customization and Tuning*).

Writer The name of the user ID associated with the WRITER task used to copy the job logs to the JES SYSOUT. If you do not specify this value, no WRITER keyword is recorded in the generated EQQOUCP sample and you can define it later in the OUCOPTS customization statement (see *Customization and Tuning*).

MLOG SWITCH

Enter Y if you plan to use the message log (MLOG) switching function which automatically switches to a second MLOG file when the first one reaches the number of records specified in `Switchlimit`. Specify Y to generate the related samples.

See also “Using two message log (MLOG) data sets” on page 42.

Switchlimit

The number of records that must be present in the MLOG file to activate the switching function. When this number is reached, the switching function activates an alternate MLOG file and archives the records of the first MLOG file into a GDG data set. You can also use the OPCOPTS SWITCHMLOGLIM parameter to specify or modify this number after the product is installed.

Note that unless you specify a positive number of records, the function is not activated, even if you specified Y in the previous field. The maximum possible value is 999999999 records.

See also “Using two message log (MLOG) data sets” on page 42.

EXTENDED AUDITING

Specify YES to activate the extended auditing feature. This value is meaningful only if you have set AMOUNT(EXTENDED) in the AUDIT statement of the controller (for details, see *Customization and Tuning*). By activating the extended auditing feature, the EQQPCS14 sample creates the EQQDBnn, EQQDBARC, and EQQDBOUT data sets. EQQDBnn and EQQDBARC are added to the controller started-task. EQQDBARC and EQQDBOUT are added to the DP batch job.

STEP AWARENESS

Specify YES to have the tracker generate a list of the step events related to the jobs you run, and send it to the primary controller. The primary controller logs the step list, without storing it. If a backup controller exists, the primary controller sends the list to the backup controller, where the information is stored.

Store on primary controller

Specify YES to store the list of step events also on the primary controller. This option can affect the controller performance.

6. Press Enter. The following panel is displayed:

```

EQQJOBSD ----- CREATE SAMPLE JOB JCL -----
Command ==>

BKPT configuration          N      (Y= Yes ,N= No)
Backup data set name      ==> TWSZ.INST.BKPT_____

BKPTOPTS statement procedure names:

DUMP Procedures for:
NCP, NCX, NXD Files      ==> _____
CX, CP1, XD1 Files       ==> _____
CX, CP2, XD2 Files       ==> _____
LTP File                 ==> _____

RESTORE Procedures:
NCP, NCX, NXD Files      ==> _____
CX, CP1, XD1 Files       ==> _____
CX, CP2, XD2 Files       ==> _____
LTP File                 ==> _____

Press ENTER to create sample job JCL.

```

Figure 18. EQQJOBSD - Create sample job JCL

BKPT configuration

Specify Y to enable the backup controller configuration.

Backup data set name

Required input. The name of the backup data set.

DUMP Procedures for NCP, NCX, NXD Files

The name of the dump procedure for NCP, NCX, and NXD files. If you do not specify any value, the default EQQSENCP is used.

DUMP Procedures for CX, CP1, XD1 Files

The name of the dump procedure for CX, CP1, XD1, and ST files. If you do not specify any value, the default EQQSECP1 is used.

DUMP Procedures for CX, CP2, XD2 File

The name of the dump procedure for CX, CP2, XD2, and ST files. If you do not specify any value, the default EQQSECP2 is used.

DUMP Procedures for LTP Files

The name of the dump procedure for the LTP file. If you do not specify any value, the default EQQSENLT is used.

RESTORE Procedures for NCP, NCX, NXD Files

The name of the restore procedure for NCP, NCX, and NXD files. If you do not specify any value, the default EQQRENCNCP is used.

RESTORE Procedures for CX, CP1, XD1 Files

The name of the restore procedure for CX, CP1, XD1, and ST files. If you do not specify any value, the default EQQRECP1 is used.

RESTORE Procedures for CX, CP2, XD2 Files

The name of the restore procedure for CX, CP2, XD2, and ST files. If you do not specify any value, the default EQQRECP2 is used.

RESTORE Procedures for LTP File

The name of the restore procedure for the LTP file. If you do not specify any value, the default EQQRESLT is used.

7. Press Enter when you have entered the information on panel EQQJOBSD. The dialog process now generates several members in the output library that you

specified. These members, which are described in Table 19, will be used at various stages in the installation.

Table 19. Sample JCL generated by the EQQJOBS dialog

Member	Description of job
EQQ9SM01	Updates the RACF router table (ICHRFR01).
EQQ9SMDE	Updates the RACF class-descriptor table (ICHRRCDE).
EQQAUDIB	Sample to invoke EQQAUDIT in batch mode outside of the dialog. Note: EQQAUDIB is used only if you specified a value for the EQQTROUT dsname and the EQQAUDIT output dsn fields in the EQQJOBSA panel.
EQQBENCR	Sample JCL to run the utility that encrypts the Windows passwords set in the USRPSW parameter of the USRREC statements.
EQQBSCAN	Uses the batch loader to scan an Application Description.
EQQBSUBS	Uses the batch loader to create the Application Descriptions and Operator Instructions.
EQQBVSAM	Deletes and defines an Application Description data set and creates an Application Description and Operator Instructions, by using the batch loader.
EQQCHKEV	Utility that checks if all events in EQQTWSIN and EQQTWSOU were correctly processed.
EQQCONOP	Sample initial parameters for the controller only.
EQQCONO	Sample started task procedure for the controller only.
EQQCONP	Sample initial parameters for a controller and tracker in same address space.
EQQCON	Sample started task procedure for a controller and tracker in same address space.
EQQDBENC	Contains the JCL to encrypt the password in the DBOPT statement.
EQQDBOPT	Sample DBOPT statement.
EQQDPCOP	JCL and usage notes for copy VSAM function.
EQQE2EP	Sample initial parameters for server and batch to define if the end-to-end scheduling with fault tolerance capabilities is active.
EQQFLWAT	Sample JCL to call filewatch utility to monitor HFS or ZFS file changes.
EQQICNVH	Sample job to migrate history DB2 tables.
EQQICNVS	Migrates VSAM files.
EQQJES2	Assembles and link-edits the JES2 EXIT7.
EQQJER2U	Restores the EXIT7 as a JES2 usermod.
EQQJER2V	Restores the EXIT5 as a JES2 usermod.
EQQJER3U	Restores the EQQUX191 and EQQUX291 as JES3 usermods.
EQQJER3Z	Restore the EQQUX291 as JES3 usermod.
EQQJES21	Assembles and link-edits the JES2 EXIT51.
EQQJES2U	Installs the JES2 EXIT7 usermod.
EQQJES2V	Installs the JES2 EXIT51 usermod.
EQQJES3	Assembles and link-edits a JES3 exit.
EQQJES3U	Installs the JES3 usermod.
EQQJES3Z	Installs the EQQUX291 as JES3 usermod.

Table 19. Sample JCL generated by the EQQJOBS dialog (continued)

Member	Description of job
EQQMTWSO	Migrates TWSOU data set size for extended job names from lrecl 120 to lrecl 160.
EQQORST	Resets the USS environment for the end-to-end scheduling with fault tolerance capabilities.
EQQOUC	Sample started task procedure for the Output collector.
EQQOUCH	Sample header template for job logs.
EQQOUCP	Sample initial parameters for the Output collector.
EQQPCS01	Allocates unique data sets within the sysplex.
EQQPCS02	Allocates non-unique data sets.
EQQPCS03	Generates a job that allocates VSAM copy data set.
EQQPCS05	Allocates files used by a controller to enable fault-tolerant workstations.
EQQPCS06	Allocates VSAM data sets for integration with the end-to-end scheduling with fault tolerance capabilities.
EQQPCS07	Allocates VSAM data sets for Restart and Cleanup.
EQQPCS08	Allocates USS files for Java utilities enablement.
EQQPCS09	Allocates the GDG root and VSAM data set used as input by the archiving process supporting the Dynamic Workload Console reporting feature.
EQQPCS10	Creates the SSL work directory used for TCP/IP communication with the controller.
EQQPCS11	Allocates data sets (EQQOUCV and EQQOUCP) used for the retrieval of job logs in the z-centric environment with the Output collector.
EQQPCS12	Allocates the GDG root to archive the MLOG files.
EQQPCS13	Allocates the GDG root to send and restore procedures.
EQQRECP1	Restore procedure for CX, CP1, XD1, ST files.
EQQRECP2	Restore procedure for CX, CP2, XD2, ST files.
EQQRENCP	Restore procedure for NCP, NCX, NXD files.
EQQRESLT	Restore procedure for LTP files.
EQQREPRO	Is invoked by EQQSMLOG to copy the contents of the outgoing MLOG file onto the GDG data set. You must copy this sample to the PARMLIB of the controller.
EQQRAD	Restore procedure for the Application Description.
EQQRJS1	Restore procedure for the primary JCL repository.
EQQRJS2	Restore procedure for the secondary JCL repository.
EQQROI	Restore procedure for the Operator Instructions.
EQQRRD	Restore procedure for the Resource Description.
EQQRSI	Restore procedure for the Side Information file.
EQQRWS	Restore procedure for the Workstation Description.
EQQSAD	Send procedure for the Application Description.
EQQSAMPI	Copies sample databases from the sample library to VSAM data sets.
EQQSECP1	Dump procedure for CX, CP1, XD1, ST files.
EQQSECP2	Dump procedure for CX, CP2, XD2, ST files.

Table 19. Sample JCL generated by the EQQJOBS dialog (continued)

Member	Description of job
EQQSENC	Dump procedure for NCP, NCX, NXD files.
EQQSENLTP	Dump procedure for LTP files.
EQQSERP	Sample initial parameters for a Server.
EQQSER	Sample started task procedure for a Server.
EQQSJS1	Send procedure for the primary JCL repository.
EQQSJS2	Send procedure for the secondary JCL repository.
EQQSLCHK	JCL to perform a syntactic check on SCRIPT library members.
EQQSMF	Updates SMF exits for IBM Workload Scheduler for z/OS.
EQQSMLOG	Sample procedure that creates the GDG data set where the outgoing MLOG file is archived when the MLOG switching function takes effect. Uses the EQQREPRO input parameter.
EQQSOI	Send procedure for the Operator Instructions.
EQQSRD	Send procedure for the Resource Description.
EQQSSI	Send procedure for the Side Information file.
EQQSW	Send procedure for the Workstation Description.
EQQTRA	Sample started task procedure for a tracker.
EQQTRAP	Sample initial parameters for a tracker.
EQQTROPT	Sample TRGOPT statement.

Generating batch-job skeletons

To ensure that all files are correctly allocated, you must generate the batch-job skeletons before creating the sample job JCL.

Several controller functions, such as daily planning, are performed by batch jobs that are submitted from the IBM Workload Scheduler for z/OS dialog. To generate the skeleton JCL for these jobs:

1. Select option 2 from the EQQJOBS main menu. This panel is displayed:

```

EQQJOBS1 ----- GENERATE IWSz BATCH-JOB SKELETONS -----
Command ==>

Enter the name of the output library. This should be a fully qualified
data set name without any enclosing apostrophes. This library should be
allocated to ISPSLIB.

Batch-job skeletons   ==> CCOPC.OPCA.JCLSKELS_____

The following data set names are used by one or more of the generated job
You can specify an asterisk (*) to indicate the name of the subsystem.

Message library name ==>
OPC.SEQMSG0_____
Parameter library   ==>
CCOPC.*.PARM_____
Member in parameter
  library           ==> BATCHOPT
Checkpoint data set ==>
CCOPC.*.CKPT_____

Press ENTER to continue

```

Figure 19. EQQJOBS1 - Generate IBM Workload Scheduler for z/OS batch-job skeletons

Batch-job skeletons

Required input. Enter the name of the library where the JCL skeletons are to be stored. Before you use the IBM Workload Scheduler for z/OS dialog to submit batch jobs, allocate this library to the ISPSLIB DD statement in the TSO session of dialog users.

“The ISPF environment” on page 188 explains how you set up the dialog. You can create a new library for the skeleton JCL members or put them in an existing skeleton-JCL library.

In the following fields, you can enter &XOPCNM. as one of the qualifiers for the data set names. This is an ISPF variable that is stored in the profile and is the same variable that you specify in option 0.1 (SUBSYSTEM NAME) in the IBM Workload Scheduler for z/OS dialogs. When a skeleton is then used by a dialog of the scheduler, &XOPCNM. is substituted with the name of the scheduler subsystem that is being used.

Ensure that &XOPCNM. ends with a period if it is not the low-level qualifier. For example, you could enter CCOPC.&XOPCNM..PARMLIB but CCOPC.&XOPCNM.PARMLIB results in a JCL error.

If you enter an asterisk (*) as a data set qualifier, the generated skeletons will contain &XOPCNM. in place of the asterisk.

Message library name

Required input. Enter the name of the library that contains the IBM Workload Scheduler for z/OS messages (SMP/E target DD name SEQMSG0).

Parameter library

Required input. Enter the name of the library that will contain the initialization statements.

Member in parameter library

Required input. Enter the name of a member in the parameter library that will contain the BATCHOPT initialization statement. The IBM Workload Scheduler for z/OS batch jobs will use this member. If you have not already created the BATCHOPT statement, you can still

generate the batch skeletons, but remember to create a member with the same name when you create the initialization statements.

Checkpoint data set

Required input. Enter the name of the checkpoint data set.

2. Press Enter, and this panel is displayed:

```

EQQJOBS2 ----- GENERATE IWSz BATCH-JOB SKELETONS -----
Command ==>

Enter the following required job stream parameters:
Non-VSAM dsn prefix ==> CCOPC.*_____

VSAM dsn prefix      ==> CCOPC.*V_____
Unit name            ==> 3390_____      Default unit name
Unit name (temp ds) ==> SYSDA_____      Unit name for temporary data sets
Unit name (sort ds) ==> SYSDA_____      Unit name for sort work data sets
SYSOUT class         ==> *_____        SYSOUT class for reports

The following information is optional:

STEPLIB dsname       ==> _____
OPC.SEQQLMD0         ==> _____
STEPDAT dsname       ==> _____

EQQMLOG dsname       ==> CCOPC.*.MLOGBAT
_____

The following information is REQUIRED WITH DBCS support:

KJSRTBL dsname       ==> _____
_____

Press ENTER to generate OPC batch-job skeletons

```

Figure 20. EQQJOBS2 - Generate IBM Workload Scheduler for z/OS batch-job skeletons

Non-VSAM dsn prefix

Required input. Enter the qualifiers that prefix the non-VSAM data set names. IBM Workload Scheduler for z/OS adds a low-level qualifier to the prefix to uniquely identify the non-VSAM data sets. For example, it adds JTARC for the job-tracking archive data set. If the subsystem name is OPCA, the data set name will be CCOPC.OPCA.JTARC when the skeleton is used by the dialogs.

VSAM dsn prefix

Required input. Enter the qualifiers that prefix the VSAM data set names. IBM Workload Scheduler for z/OS adds a low-level qualifier to the prefix to uniquely identify each IBM Workload Scheduler for z/OS VSAM data set. For example, it adds WS for the workstation description data set. If the subsystem name is OPCA, the data set name will be CCOPC.OPCAV.WS when the skeleton is used by the dialogs.

Unit name

Required input. Enter a device name that is valid at your installation. This can be a device type, for example 3380, or a group name, for example PROD or TEST.

Unit name (temp ds)

Required input. Enter a device name that can be used for temporary data sets.

Unit name (sort ds)

Required input. Enter a device name that can be used for sort-work data sets.

SYSOUT class

Required input. Specify the SYSOUT class that you want to use for the reports that are generated by the batch jobs.

STEPLIB dsname

Optional. Enter the name of the IBM Workload Scheduler for z/OS load module library if the load modules are not in a data set included in an active LNKLST member.

STEPCAT dsname

Optional. Enter the name of a private catalog if one or more data sets cannot be reached via the master catalog. To customize the EQQAUDNS skeleton clist with the appropriate loadlib that should be referenced when audit/debug is invoked, you must specify the dsname.

EQQMLOG dsname

Optional. Enter the name of a message log data set if messages are not sent to SYSOUT. This must not be the same data set that is used by a tracker, controller, or standby controller.

KJSRTBL dsname

Required if you use the Japanese language feature. Enter the name of a data set that will be used when sorting fields containing DBCS data.

3. Press Enter. The following panel is displayed:

```

EQQJOBSA ----- GENERATE IWSz BATCH-JOB SKELETONS -----
Command ==>
Specify if you want to use the following optional features:

END-TO-END WITH FAULT TOLERANCE:      N   (Y= Yes ,N= No)
(To schedule jobs on fault-tolerant workstations)

RESTART AND CLEAN UP (DATA STORE):    Y   (Y= Yes ,N= No)
(To be able to retrieve joblog,
execute data set clean up actions and step restart)

FORMATTED REPORT OF TRACKLOG EVENTS:  N   (Y= Yes ,N= No)
EQQTROUT dsname      ==> _____
EQQAUDIT output dsn  ==> _____

JAVA UTILITIES ENABLEMENT:            N   (Y= Yes ,N= No)
Work Directory       ==> /var/TWS/inst_____
                    ==> _____
                    ==> _____

JZOS PDSE Library   ==> _____
JZOS Load Module Name ==> JVMLDM66
REXX SYSEXEC dsname ==> OPC.SEQQMISC
Input XML dsname for ==> TWS.EVLIB.XML($$$$$$$)_____
data set triggering

Press ENTER to generate OPC batch-job skeletons

```

Figure 21. EQQJOBSA - Generate IBM Workload Scheduler for z/OS batch-job skeletons

END-TO-END WITH FAULT TOLERANCE

Specify Y if you want to work with IBM Workload Scheduler fault-tolerant workstations.

RESTART AND CLEAN UP (Data Store)

Specify Y if you want to use the Restart and Cleanup feature.

FORMATTED REPORT OF TRACKING EVENTS

Specify Y if you want to use the feature that produces a formatted report of the tracklog events.

EQQTROUT dsname

This entry is optional. Specify the name of the data set in which DP Extend and Replan writes tracklog events. Leave blank if you want the corresponding DD card for these jobs to specify DUMMY as in previous releases. Type out if you plan to use sample EQQAUDIB (see Table 19 on page 67).

EQQAUDIT output dsn

Specify the name of a data set where the EQQAUDIT output is to be written. Required if FORMATTED REPORT OF TRACKLOG EVENTS is set to Y.

Work Directory

Specify the directory where the subsystem specific HFS or ZFS files are stored. Each subsystem supporting the JAVA utility must have its own work directory.

JZOS PDSE Library

JZOS PDSE Library Specify the PDSE containing the JZOS Batch Launcher JVMLDM module.

JZOS Load Module Name

Specify JVMLDM66, that is the JZOS Batch Launcher load module name used with 64-bit SDK for z/OS 6.0.

REXX SYSEXEC dsname

Specify the installation SEQQMISC library containing the REXX programs EQQRXARC and EQQRXTRG.

Input XML dsname for data set triggering

Specify the name of the input data set containing the event rules in XML format used to produce the data set triggering configuration files that will be sent to trackers. The default data set is provided in the panel.

Note: In controller MLOG dsn, EQQTROUT dsname, and EQQAUDIT output dsn you can use an asterisk (*) for the subsystem name. It will be replaced with the current subsystem name when the dialog is invoked.

4. Press Enter when you have entered the information on panel EQQJOBSA. The dialog now generates the batch-job skeleton members.

After completing this procedure, you can proceed with the creation of the sample job JCL as described in "Creating the sample job JCL" on page 58.

If you are not sure at this stage what some of the values will be, it does not matter. You can rerun the dialog as many times as you want to regenerate the skeletons. You can also edit the generated skeletons manually.

This table shows the JCL skeleton members that EQQJOBS generates:

Table 20. Controller skeleton JCL generated by the EQQJOBS dialog

Member	Batch job description
EQQADCDS	Application cross-reference of conditional dependencies.
EQQADCOS	Calculate and print run dates of an application.
EQQADDES	Application cross-reference of external dependencies.

Table 20. Controller skeleton JCL generated by the EQQJOBS dialog (continued)

Member	Batch job description
EQQADPRS	Application print program.
EQQADXRS	Application cross-reference program.
EQQADX1S	Application cross-reference of selected fields.
EQQAMUPS	Application description mass update.
EQQAPARS	Procedure to gather diagnostic information.
EQQAUDIS	Extract and format job tracking events (batch invocation).
EQQAUDNS	Extract and format job tracking events (interactive invocation) Note: Ensure to copy this member from the library where it was created by EQQJOBS into a procedure library. This step is required since this member must be invoked interactively.
EQQDBARS	Daily Planning - Historical run data archiver for Dynamic Workload Console reporting feature.
EQQDPEXS	Daily planning - plan next period.
EQQDPPRS	Daily planning - print current period results.
EQQDPRCS	Daily planning - replan current period.
EQQDPSJS	Daily planning -DBCS sort step.
EQQDPSTS	Daily planning - normal sort step.
EQQDPTRS	Daily planning - plan a trial period.
EQQJVPRS	Print JCL variable tables.
EQQLEXTS	Long-term planning - extend the long-term plan.
EQQLMOAS	Long-term planning - modify all occurrences.
EQQLMOOS	Long-term planning - modify one occurrence.
EQQLPRAS	Long-term planning - print all occurrences.
EQQLPRTS	Long-term planning - print one occurrence.
EQQLTRES	Long-term planning - create the long-term plan.
EQQLTRYs	Long-term planning - trial.
EQQOIBAS	Operator instructions - batch program.
EQQOIBLS	Operator instructions - batch input from a sequential data set.
EQQSYRES	Daily Planning - Symphony Renew.
EQQTPRPS	Print periods.
EQQTPRTS	Print calendars.
EQQTRBLS	Event driven workload automation - Create configuration files for data set triggering
EQQWPRTS	Print workstation descriptions.

Generating Data Store samples

To create the Data Store samples:

1. Select option 3 from the EQQJOBS application menu. The following panel is displayed:

```

EQQJOBS5 ----- CREATE DATA STORE SAMPLES -----
Command ==>

The data set names specified on this panel should be fully qualified
names without any enclosing apostrophes.

Enter the name of the output library:

Sample job JCL      ==> CCOPC.OPCA.JCLDS _____

Job statement information:

==> //SYSPROG1 JOB (111111,2222),'OPCESA BATCH',CLASS=B,MSGCLASS=H,___
==> //          MSGLEVEL=(1,1),NOTIFY=SYSPROG _____
==> _____
==> _____

The following data set names are used by one or more of the generated jobs.
Message library name ==> OPC.SEQMSG0 _____
Parameter library   ==> CCOPC.OPCEDS.PARM _____

Press ENTER to continue

```

Figure 22. EQQJOBS5 - Create Data Store samples

Sample job JCL

Required input. Enter the name of the library to which you want the generated Data Store samples written. The library must be allocated before you generate the Data Store samples. Ensure that the library that you specify has sufficient directory blocks to store all the sample members that are generated by EQQJOBS (see Table 21 on page 78).

Job statement information

Required input. Enter a JOB statement that follows standard JCL syntax and your installation standards.

Message library name

Required input. Enter the name of the library that contains the scheduler messages (SMP/E target DDNAME SEQMSG0).

Parameter library

Required input. Enter the name of a library that will contain the initialization statements. This library must be allocated by the user. Use a name different from that of the controller and tracker parameter library.

2. Press Enter, and this panel is displayed:

```

EQQJOBS6 ----- CREATE DATA STORE SAMPLES -----
Command ==>

Enter the following required job stream parameters:
Non-VSAM dsn prefix ==> CCOPC.OPCA _____
VSAM dsn prefix     ==> CCOPC.OPCAV _____
Unit name           ==> 3390 _____ Default unit name
Primary volume serial ==> PROD01 _____ Primary volume serial for VSAM

The following information is optional:
STEPLIB dsname      ==> OPC.SEQQLMD0 _____
VSAMCAT dsname      ==> _____
VSAM password       ==> _____

Press ENTER to continue

```

Figure 23. EQQJOBS6 - Create Data Store samples

Non-VSAM dsn prefix

Required input. Enter the qualifiers that prefix the non-VSAM data set names. The scheduler adds a low-level qualifier to the prefix to uniquely identify each Data Store non-VSAM data set.

VSAM dsn prefix

Required input. Enter the qualifiers that prefix the VSAM data set names. The scheduler adds a low-level qualifier to the prefix to uniquely identify each Data Store VSAM data set.

Unit name

Required input. Enter a device name that is valid at your installation. This could be a device type, for example 3390, or a group name, for example PROD or TEST.

Primary volume serial

Required input. Enter a volume that will be used by sample job EQQPCS04 to allocate the primary data sets.

STEPLIB dsname

Optional. Enter the name of the scheduler load module library if the load modules are not in a data set included in an active LNKLIST member.

VSAMCAT dsname

Optional. Enter the name of a catalog in which VSAM data sets are to be defined if they are not to be defined in the master catalog.

VSAM password

Optional. Enter the password for the VSAM catalog if the catalog is password-protected.

3. Press Enter, and this panel is displayed:

```

EQQJOBS7 ----- CREATE DATA STORE SAMPLES -----
Command ==>

Enter the parameters to build DSTOPTS and DSTUTIL options samples:

Reserved destination      ==> OPCX
Connection type           ==> SNA (SNA/XCF/TCP)
  SNA Data Store luname   ==> I9PC45AA (only for SNA connection)
  SNA Controller luname   ==> I9PC45RA (only for SNA connection)
  Xcf Group               ==> _____ (only for XCF connection)
  Xcf Data Store member   ==> _____ (only for XCF connection)
  Xcf FL task member      ==> _____ (only for XCF connection)
  TCP Controller host name: _____ (only for TCP connection)
  ==> _____
  TCP Controller port number ==> _____ (only for TCP connection)
Jobdata ret. period       ==> 2_ (number of days)
JobLog retrieval          ==> Y (Y/N)
  Max n. lines to store   ==> 0
  JobLog ret. period      ==> 5_ (number of days)

Press ENTER to create sample job JCL

```

Figure 24. EQQJOBS7 - Create Data Store samples

Reserved destination

Required input. Enter the Data Store reserved output destination. It must correspond to DSTDEST parameter in RCLOPTS option.

Connection Type

Required input. Enter the connection method used to handle communication between FN/FL tasks and Data Store. It can be SNA or XCF.

SNA Data Store luname

Enter Data Store VTAM application name if SNA connection type has been chosen.

SNA Controller luname

Enter controller FN task VTAM application name if SNA connection type has been chosen.

Xcf Group

Enter the name of XCF group if XCF connection type has been chosen.

Xcf Data Store member

Enter the name of Data Store XCF member if XCF connection type has been chosen.

Xcf FL task member

Enter the name of FL task XCF member if XCF connection type has been chosen.

Job data retention period

Enter the Data Store structured information retention period. It consists of the interval in days used by the online cleanup and is necessary to be able to use the Restart and Cleanup feature.

Joblog retrieval

Specify if the joblog retrieval must be enabled. This means that the Data Store will save the unstructured data in the joblog.

Max n. of lines to store

Enter the maximum number of user sysout lines to be stored. The range is 0 to 10000.

Joblog retention period

Enter the Data Store unstructured information retention period. It consists of the interval in days used by the online cleanup and is necessary to enable the Joblog Browse function.

TCP Controller host name

Specify the Controller TCP/IP host name if you chose TCP in Connection type.

TCP Controller port number

Specify the Controller TCP/IP port number if you chose TCP in Connection type.

4. Press Enter when you have entered the information on panel EQQJOBS7. The dialog now generates several members in the output library that you specified. These members, which are described in Table 21 is used at various stages in the installation:

Table 21. Data Store samples generated by the EQQJOBS dialog

Member	Sample Description
EQQCLEAN	Sample procedure invoking EQQCLEAN program
EQQDSCL	Batch Clean Up sample
EQQDSCLP	Batch Clean up sample parameters
EQQDSEX	Batch Export sample
EQQDSEXP	Batch Export sample parameters
EQQDSIM	Batch Import sample
EQQDSIMP	Batch Import sample parameters
EQQDSRG	Batch sample reorg
EQQDSRI	Batch Recovery index
EQQDSRIP	Batch Recovery index parameters
EQQDST	Sample procedure to start Data Store
EQQDSTP	Parameters for sample procedure to start Data Store
EQQPCS04	Allocate VSAM data sets for Data Store

Creating the Workload Automation Programming Language samples

To create the Workload Automation Programming Language samples:

1. Select option 4 from the EQQJOBS application menu. The following panel is displayed:

ISPF Messages

The ISPF messages library.

ISPF Panels

The ISPF panels library.

ISPF Skeletons

The ISPF skeletons library.

ISPF Tables

The ISPF tables library.

Subsys

The IBM Workload Scheduler for z/OS subsystem.

Language

The Workload Automation Programming Language language. Only English (EN) is supported.

Version

The IBM Workload Scheduler for z/OS version.

Table 22. Workload Automation Programming Language samples generated by the EQQJOBS dialog

Member	Sample Description
EQQYXJCL	Workload Automation Programming Language for z/OS batch procedure.
EQQWCMD1	Use only on a started-task (STC) workstation to run Workload Automation Programming Language commands for an IBM Workload Scheduler for z/OS system configured with OPCOPTS RCLEANUP(YES).
EQQWCMD2	Use only on a started-task (STC) workstation to run Workload Automation Programming Language commands for an IBM Workload Scheduler for z/OS system configured with OPCOPTS RCLEANUP(NO).
EQQWTSO1	Sets up the Workload Automation Programming Language environment for TSO, runs Workload Automation Programming Language commands, processes the result, and resets the environment.
EQQWTSO2	This sample assumes that the Workload Automation Programming Language environment is already set up in the TSO LOGON procedure or startup REXX/CLIST. It runs commands and then processes the results.
EQQWTSO3	Sets up the Workload Automation Programming Language environment for TSO, to be called by another REXX script that sets Workload Automation Programming Language commands and then processes the results.
EQQWTSO4	This sample assumes that the Workload Automation Programming Language environment is already setup in the TSO LOGON procedure or startup REXX/CLIST. It is called by another REXX that sets Workload Automation Programming Language commands and then processes the results.
EQQWTSX3	Calls another REXX script that sets up the Workload Automation Programming Language environment to run the commands specified here.
EQQWTSX4	Calls another REXX script that assumes that the Workload Automation Programming Language environment has already been set up in the TSO LOGON procedure or startup CLIST. Then it runs the commands specified here.

Step 5. Adding SMF and JES exits for event tracking

Perform this task if you are installing a tracker.

IBM Workload Scheduler for z/OS tracks the progress of jobs and started tasks through the z/OS system by using JES and SMF exit points. Add all these exits on each z/OS system where you will start IBM Workload Scheduler for z/OS.

To simplify the installation of IBM Workload Scheduler for z/OS event tracking, several sample event-tracking exits can be found in your sample library, SEQQSAMP. To assemble and install exits, you can use the sample JCL provided to install the exits as SMP/E *usermods* or alternatively you can assemble and link-edit the exits yourself. For JES exits, apply *usermods* in the CSI where JES is included: this is the best method. It has the advantage that SMP automatically reassembles the exits if maintenance is applied to the JES control blocks that IBM Workload Scheduler for z/OS is dependent on.

If you install a new release of IBM Workload Scheduler for z/OS in a new CSI, and the JES usermod is already installed in the same CSI as a previous release, follow these steps:

1. Apply any necessary tolerance PTFs so that the previous release can run with the new exit code.
2. Change the DDDEFs for JES so that they point to the SEQQSAMP and SEQQMAC0 libraries of the *new* release.
3. APPLY REDO the JES usermod. This reassembles the exits with the new code.

Note: The exits that you install on IBM Workload Scheduler for z/OS V9.3 are compatible with earlier versions of the product. Nevertheless, if you are using JES2 ensure that you have installed APAR PI24927 on IBM Workload Scheduler for z/OS systems earlier than V9.3.

The sample exits all use the EQQEXIT macro to create event-generating code. For more information on the EQQEXIT macro, see Appendix C, “Invoking the EQQEXIT macro,” on page 387.

Table 23 describes the samples that you can use to generate and install the exits. The sample exit, skeleton JCL, and *usermod* entries identify members of the SEQQSAMP library. The event types in the table are prefixed with A for JES2 or B for JES3, when they are created by the exit. (See “Verifying tracking events” on page 163 for more information about event types.)

Table 23. Sample exits for IBM Workload Scheduler for z/OS

Exit name	Exit type	Sample exit	Sample JCL/ usermod	Event supported	Event type
IEFACTRT	SMF	EQQACTR1	EQQSMF	Job and step completion	3J,3S
IEFUJI	SMF	EQQUJI1	EQQSMF	Job start	2
IEFU83	SMF	EQQU831	EQQSMF	End of print group, purge (JES3 only), data set triggering support, and automatic change support	4,5,S,T
EXIT7	JES2	EQQXIT74	EQQJES2/ EQQJES2U	JCT I/O exit for JES2, purge	1,3P,5
EXIT51	JES2	EQQXIT51	EQQJES21/ EQQJES2V	JES2 QMOD phase change exit, z/OS 1.7 and later	1

Table 23. Sample exits for IBM Workload Scheduler for z/OS (continued)

Exit name	Exit type	Sample exit	Sample JCL/ usermod	Event supported	Event type
IATUX19	JES3	EQQUX191	EQQJES3/ EQQJES3U	Output processing complete	3P
IATUX29	JES3	EQQUX291	EQQJES3/ EQQJES3U	On job queue	1

SMF only

The EQQU831 sample generates type 4 and type 5 events and also generates resource availability events when a data set is closed after read or update processing. See “Implementing support for data set triggering” on page 92 for more information.

You must tailor the sample JCL to the requirements of your installation. If you have already run EQQJOBS, tailored versions of the JCL will already exist in the EQQJOBS output library. Alternatively, you can copy any of the members from the SEQQSAMP library to one of your own libraries and manually tailor the JCL.

If you are unfamiliar with how to activate SMF exits, see “Updating SMF parameters” on page 86 and the documentation for SMF.

JES2 only

The load module names are the same as the exit names, except for JES2. The load module of the JES2 exits, which are EXIT7 and EXIT51, are called OPCAXIT7 and TWSXIT51, and their entry points are called OPCAENT7 and TWSSENT51, respectively.

If your z/OS system is a JES2 system, include these records in the JES2 initialization member:

JES2 Initialization Statements

```
LOAD(OPCAXIT7) /*
Load IBM Workload Scheduler for z/OS exit mod */
EXIT(7) ROUTINES=OPCAENT7,STATUS=ENABLED /*
Define EXIT7 entry point */
```

If your z/OS system is version 1.7 or later, add the following records to the JES2 initialization member:

```
LOAD(TWSXIT51) /*
Load IBM Workload Scheduler for z/OS exit mod */
EXIT(51) ROUTINES=TWSSENT51,STATUS=ENABLED /*
Define EXIT51 entry point */
```

To dynamically install the JES2 exits for IBM Workload Scheduler for z/OS, use these commands once the modules are available in the LNKLIST:

```
$ADD LOADMOD(OPCAXIT7),STORAGE=PVT
$T EXIT(7),ROUTINES=OPCAENT7,
STATUS=ENABLED
$ADD LOADMOD(TWSXIT51),STORAGE=PVT
$T EXIT(51),ROUTINES=TWSSENT51,
STATUS=ENABLED
```

To put a new version of an exit (that was previously installed) in place, use these commands once the modules are available in the LNKLIST:

```
$TLOADMOD(OPCAXIT7),REFRESH  
$TLOADMOD(TWSXIT51),REFRESH
```

For more information on JES2 initialization statements, see *JES2 Initialization and Tuning Reference*.

JES3 only

To activate the exits for a JES3 system, you can link them to a library that is concatenated ahead of SYS1.JES3LIB. Alternatively, you can replace the existing exits in SYS1.JES3LIB with the IBM Workload Scheduler for z/OS-supplied IATUX19 and IATUX29 exits. For more information, see *JES3 Initialization and Tuning Reference*. If you get RC=4 and the warning ASMA303W Multiple address resolutions may result when you assemble IATUX19 running the EQQJES3/EQQJES3U sample, you can ignore the message. If version ASMA90 of the compiler reports errors, and the RMODE=ANY statement is defined, remove the RMODE=ANY statement from the sample exit.

Step 6. Updating SYS1.PARMLIB

The following sections describe the updates to SYS1.PARMLIB for your environment:

Defining subsystems

When you define the subsystem names of the IBM Workload Scheduler for z/OS controllers and trackers, consider the following:

- The Subsystem/STC name of IBM Workload Scheduler for z/OS controllers is unique within the PLEX. If two different controllers (regardless of their location) are configured to track work on the same z/OS system, they must have different Subsystem/STC names.
- Because subsystem names on a given LPAR must be unique, and because all IBM Workload Scheduler for z/OS trackers and controllers started tasks must have the same name as their associated subsystems, all started tasks on any given LPAR must have unique names. That is, inside an MVS image, controllers and trackers must have unique Subsystem/STC names.
- Trackers running on different LPARs but connected to the same controller can have the same Subsystem/STC name. In this case, system variables like &SYSNAME can be used with the condition that each tracker uses different IBM Workload Scheduler for z/OS data sets. The tracker name cannot be the same as the name of a controller.

You must define the name of every new IBM Workload Scheduler for z/OS subsystem in the active subsystem-name-table member of SYS1.PARMLIB. Install at least two IBM Workload Scheduler for z/OS controlling systems, one for testing and one for your production environment.

Note: It is recommended that you install the tracker and the controller in separate address spaces on the controlling system.

To define the subsystems, update the active IEFSSN nnn member in SYS1.PARMLIB. Include records as in the following example:

```
Subsystem definition record  
SUBSYS SUBNAME(subsystem name) INITRTN(module name) INITPARM ('maxecsa,suffix')
```

subsystem name

The name assigned to an IBM Workload Scheduler for z/OS subsystem. The name must be from 2 to 4 characters. All the subsystem names, as defined in the SYS1.PARMLIB member IEFSSN*nn*, must be unique within a GRS complex with the exception of a standby controller. Also, the subsystem names of the controllers must be unique within your OPCplex/OPC network, both local and remote systems. IBM Workload Scheduler for z/OS requires the started task name or job name used for an IBM Workload Scheduler for z/OS address space to exactly match the name of the associated subsystem.

module name

The name of the subsystem initialization module, EQQINITM.

maxecsa

Defines the maximum amount of extended common service area (ECSA) that is used to queue job-tracking events. The value is expressed in kilobytes (1 KB equals 1024 bytes). The default is 4, which means that a maximum of 4 KB (4096 bytes) of ECSA storage is needed to queue job-tracking events. The maximum value allowed for MAXECSA is 2816.

suffix The module name suffix for the EQQSSCM module that EQQINITM loads into common storage. EQQSSCM is the subsystem communication module. The suffix must be a single character. Because the name of the module shipped with IBM Workload Scheduler for z/OS is EQQINITM, specify a suffix value of M. If you do not provide a suffix, EQQINITM attempts to load module name EQQSSCMM. You can also specify a subsystem communication module name in the SSCMNAME keyword of the OPCOPTS initialization statement to load an updated version of the module before a scheduled IPL. For details, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

“Updating the z/OS link-library definition” on page 88 provides more information about EQQSSCM modules.

This example illustrates a record you can include in the SYS1.PARMLIB IEFSSN*nn* member:

```
/*Subsystem definition example*/
SUBSYS SUBNAME(OPCA) INITRTN(EQQINITM) INITPARM ('100,M')
```

The record defines an IBM Workload Scheduler for z/OS subsystem called OPCA. This represents a tracker. Its initialization module is EQQINITM. The amount of ECSA that is allocated, 101104 bytes, is enough for 1136 job-tracking events. Because suffix value M is specified, EQQINITM loads module EQQSSCMM.

Calculating MAXECSA values

IBM Workload Scheduler for z/OS allocates ECSA storage for job-tracking events in blocks of 1424 bytes. Each block is equivalent to 16 events. Table 24 on page 85 gives examples of the storage needed for, the storage actually allocated, and the events accommodated for several **maxecsa** values. The number of events created for each job or started task in your environment is influenced by the definitions in the EWTROPTS initialization statement. Every job or started task creates a minimum of six events. If the job or started task generates output and PRINTEVENTS(ALL) or PRINTEVENTS(END) is specified, an event is created when each output group is purged. If STEPEVENTS(ALL) is specified, an event is created for every step in the job or started task.

If you want to calculate values that are not shown in Table 24 for a given MAXECSA value, use this method:

- Space requested = MAXECSA * 1024
- Blocks = space requested / 1424 (round down to a whole number)
- Space allocated = blocks * 1424
- Events accommodated = blocks * 16

Table 24. Examples of MAXECSA storage values

MAXECSA value	Amount of MAXECSA space requested	Blocks of ECSA space allocated (bytes)	Number of events accommodated
0	0	0 (0)	0
4	4096	2 (2848)	32
8	8192	5 (7120)	80
16	16384	11 (15664)	176
36	36864	25 (35600)	400
72	73728	51 (72624)	816
100	102400	71 (101104)	1136
200	204800	143 (203632)	2288
400	409600	287 (408688)	4592
500	512000	359 (511216)	5744

Note:

1. Allocate enough ECSA storage so that job-tracking events are not lost when the IBM Workload Scheduler for z/OS event-writer subtask is not active. When the event writer is active, the number of queued events in ECSA is almost always 0. Allocate enough ECSA for the maximum amount of time you expect the event writer to be inactive.

For example, after the IPL of a z/OS system, job-tracking events can occur before the tracker address space has become active. If you expect a maximum of 50 events to occur during this time, you should set a MAXECSA value of 8, as shown in Table 24. When the event writer becomes active, the queued events are processed and removed from ECSA.

If events are lost, message EQQZ035E is stored to the message log. For a description of this message, see *Tivoli Workload Automation: Messages and Codes*.

2. ECSA storage for job-tracking events is required only if the started task includes an event-writer subtask. On a controlling system, you can have one address space running only an event writer subtask, and another one running the controller functions and the remaining tracker functions. In this situation, you must specify a MAXECSA value of 0 for the subsystem that contains the controller functions.
3. To submit only trackers that have OPCOPTS EWTRTASK(NO) set, the MAXECSA value must also be 0. Do not define subsystems for trackers that will never be started, because this causes waste of ECSA storage. Even if performance monitors might show that a tracker has more ECSA than the value specified by the MAXECSA parameter, this extra ECSA belongs to other subsystems, and the total amount of ECSA used by IBM Workload Scheduler for z/OS for trackers never exceeds the total number of subsystems defined multiplied by their MAXECSA value.
4. All ECSA storage is allocated above the 16 MB line.

Authorizing the load-module library

You must update the active authorized-program-facility member (IEAAPF m , or PROG m) to authorize the load-module library. Each record, except the last, ends with a comma. For the following example, assume that you have installed IBM Workload Scheduler for z/OS load modules in the data set OPC.SEQQLMD0 and that this data set is on volume ABC123. To authorize this library, insert this record before the last entry in the IEAAPF m :

```
OPC.SEQQLMD0    ABC123,
```

or update the PROG m member.

Note: Libraries that are defined in the IEAAPF m or PROG m member are authorized only if they remain on the volume specified. If DFHSM is used in your system, change DFHSM parameters so that the new authorized library is not migrated by DFHSM.

Updating SMF parameters

The SMFPRM m member defines parameters for the System Management Facilities (SMF). You must verify that the active SMF parameter member, SMFPRM m , specifies that all SMF exits used by IBM Workload Scheduler for z/OS event tracking are activated, and that the required SMF records are being collected. If this is not the case, you must update the active SMF parameter member. Event tracking requires these SMF exits:

IEFUJI

Job initiation exit

IEFACTRT

Job-end and step-end exits

IEFU83

Record write exit. It is optional, and required only for data set triggering, automatic time change, and print event functions.

IBM Workload Scheduler for z/OS uses the following SMF record types:

- 6 For PRINT (A4 and B4) events, used only for tracking work on PRINT workstations
- 14 Only for data set triggering with SRREAD=YES
- 15 For data set triggering with SRREAD=YES or SRREAD=NO
- 18 Only if you want to monitor renaming data sets.
- 26 For all job tracking
- 30 For all job tracking
- 64 Only for data set triggering with VSAM data sets
- 90 Only if you want automatic daylight savings time change

IBM Workload Scheduler for z/OS requires more SMF records to be collected if you install the SMF IEFU83 exit with SRREAD set to YES on the EQQEXIT invocation. Specify this if you want special resource availability events automatically generated when a data set is closed after being opened for:

- Read processing
- Output processing
- Either read or output processing

These SMF records are needed:

- Type 14 records are required for non-VSAM data sets opened for INPUT or RDRBACK processing.
- Type 15 records are required for non-VSAM data sets opened for output.
- Type 64 records are required for VSAM data sets.
- Type 90 records support daylight savings time automatically (optional).

You can specify that the SMF records used by the exit are not written to the SMF log. If your installation does not currently collect SMF records 14, 15, or 64, but you want resource availability events automatically generated, change the EQQU831 sample so that these records are not written to the SMF log.

To avoid data set triggering, and thus to improve performance, specify SRREAD=NO in the IEFU83 SMF exit on invocation of the EQQEXIT macro. The SRREAD=NO parameter prevents data set triggering for only SMF record type 14.

Active exits are defined by the EXITS parameter of the SYS and SUBSYS keywords. An example of these keywords is:

```
/*SYS and SUBSYS keywords*/
SYS(TYPE(6,14,15,18,26,30,60,62,64,90),EXITS(IEFU83,IEFACTRT,IEFUJI))
SUBSYS(STC,EXITS(IEFUJI,IEFACTRT,IEFU83))
SUBSYS(JESn,EXITS(IEFUJI,IEFACTRT,IEFU83))
```

Note:

1. JESn is either JES2 or JES3. This parameter does not refer to JES itself, but to batch jobs handled by JES. So do not suppress exit invocation. Ensure that you do not specify TYPE6=NO and TYPE26=NO on the JOBCLASS and STCCCLASS statements of the JES2 initialization parameters.
2. You might find it useful during installation to code two SMFPRMnn members, one with the exits active and the other with the exits inactive. You can then use the SET SMF=nn z/OS command to switch your current SMF parameters to the new member. By switching back, using the SET SMF=nn command, you avoid the need to re-IPL, if you encounter a problem.
3. Exits for SUBSYS STC are required by IBM Workload Scheduler for z/OS.

Use the PROGnn parmlib member to specify installation exits and control their use. Using PROGnn, you can associate multiple exit routines with installation exits at IPL, or while the system is running. IBM recommends that you use PROGnn in addition to SMFPRMnn to specify exits, whether or not you want to take advantage of these functions.

The following example shows how you can specify SMF exits in a PROGxx parmlib member. If you specify this in SMFPRMnn:

```
SYS(...EXITS(IEFU83,IEFACTRT,IEFUJI))
```

you would add this to get the equivalent processing in PROGnn:

```
EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(IEFU83)
EXIT ADD EXITNAME(SYS.IEFACTRT) MODNAME(IEFACTRT)
EXIT ADD EXITNAME(SYS.IEFUJI) MODNAME(IEFUJI)
```

When you associate new exit routines with SMF exits through PROGnn or the SETPROG command, you must use the following naming conventions:

- For exits listed on the EXITS keyword of the SYS statement in SMFPRMnn, each exit will have the name SYS.xxxx (where xxxx is one of the exits listed).

- For exits listed on the EXITS keyword of the SUBSYS statement of SMFPRMnn, each exit will have the name SYSzzzzz.xxxx (where zzzz is the name of the subsystem and xxxx is one of the exits listed).

If you define two members in SYS1.PARMLIB with two different names, for example, PROG03 in which there is the statement EXIT ADD EXITNAME(SYS.1 EFACTRT) MODNAME(EQQACTR1), you can switch to the version EQQACTR1 without re-ipling by issuing the command:/SET PROG=03

If you are using FTP, you must add the following statement to the SMFPRMxx member:

```
SUBSYS(OMVS,EXITS(IEFUJI,IEFU83))
```

Also, these statements must be added to the PROGnn member, making sure that you replace MODNAME with the module name that was used when the exits were link-edited:

```
EXIT ADD EXITNAME(SYSOMVS.IEFU83) MODNAME(EQU831)
EXIT ADD EXITNAME(SYSOMVS.IEFUJI) MODNAME(EQUJ11)
```

For information on using PROGnn to control the use of exits and exit routines, see *z/OS Initialization and Tuning Reference*

Updating z/OS dump options

The sample JCL procedure for an IBM Workload Scheduler for z/OS address space includes a DD statement and a dump data set is allocated by the EQQPCS02 JCL created by EQQJOBS. SYSMDUMP is the dump format preferred by the service organization.

Ensure that the dump options for SYSMDUMP include RGN, LSQA, TRT, CSA, and GRSQ on systems where an IBM Workload Scheduler for z/OS address space will run. To display the current SYSMDUMP options, issue the z/OS command DISPLAY DUMP,OPTIONS. You can use the CHNGDUMP command to alter the SYSMDUMP options. Note that this command changes the parameters only until the next IPL is performed.

To dump an IBM Workload Scheduler for z/OS address space by using the z/OS DUMP command, the SDUMP options must specify RGN, LSQA, TRT, CSA, and GRSQ. Consider defining these options as your system default.

When dumping the controller address space, if JTOPTS parameter CRITJOBS(YES) was specified or taken as the default, you must also dump the data spaces; this is important if you need to collect information when a critical path problem occurs. Add DSPNAME=('ZZZZ',*) to the DUMP command, where ZZZZ is the controller subsystem name.

Updating the z/OS link-library definition

If you installed IBM Workload Scheduler for z/OS in a separate load-module library, it is recommended that you define this library in the active LNKLSTnn member. Alternatively, you can define the load-module library on the STEPLIB DD statement of the started-task JCL and TSO logon procedures of IBM Workload Scheduler for z/OS dialog users.

If you installed load modules in the data set OPC.SEQQLMD0 and this data set is cataloged in the master catalog, insert this record before the last entry in the LNKLSTnn member to add this library to the link library concatenation:

Adding LINKLIB
OPC.SEQQLMD0

If you choose not to define the IBM Workload Scheduler for z/OS load-module library in the LNKLST nn member, you *must*:

- Copy the tracker modules, EQQINITM and EQQSSCMM, to a library in the z/OS link-library concatenation. EQQINITM is used by the master-scheduler-initialization function when the z/OS system is being IPLed. EQQINITM then loads EQQSSCMM into common storage. EQQSSCMM is about 23KB and is placed above the 16MB line. Remember to copy the modules again whenever they are updated by IBM Workload Scheduler for z/OS maintenance. This is especially important for the EQQSSCMM module, which must be at the same update level as the rest of the IBM Workload Scheduler for z/OS code.
- Define the IBM Workload Scheduler for z/OS load-module library on a STEPLIB DD statement in the started-task JCL.
- Define the IBM Workload Scheduler for z/OS load-module library on a STEPLIB DD statement in the TSO logon procedure of all IBM Workload Scheduler for z/OS dialog users.
- Load the dialog module, EQQMINOM, from an APF-authorized library. If you define the IBM Workload Scheduler for z/OS load-module library on a TSO STEPLIB DD statement, and any of the other libraries defined on this DD statement are not authorized, you must copy EQQMINOM to another library in the LNKLST concatenation so that it is loaded APF authorized. You must also remember to copy the module again whenever it is updated by IBM Workload Scheduler for z/OS maintenance.

Updating XCF initialization options

This section is useful if you use XCF for communication.

XCF initialization options are specified in the COUPLE nn member of SYS1.PARMLIB. If you have not specified your own COUPLE nn member, the system uses the default member, COUPLE00. The IBM-supplied COUPLE00 member causes the system to be IPLed in XCF-LOCAL mode. This mode is *not* supported by IBM Workload Scheduler for z/OS. So ensure that your system uses a COUPLE nn member that does not IPL the system in XCF-LOCAL mode. The COUPLE nn member must include the PCOUPLE keyword of the COUPLE statement. If this is omitted, XCF is initialized in XCF-LOCAL mode. For IBM Workload Scheduler for z/OS purposes, you can use the default values for the remaining XCF options.

```
COUPLE $nn$  example
COUPLE  SYSPLEX(PLEX1)           /* SYSPLEX name           */
        PCOUPLE(PLEX1.COUPLE1)  /* Primary couple data set */
        ACOUPLE(PLEX2.COUPLE2)  /* Alternate couple data set*/
        MAXMSG(2000)            /* No of 1k message buffers */
CLASSDEF CLASS(DEFAULT)        /* Default transport class */
        CLASSLEN(956)          /* Message length         */
        GROUP(OPCGRP,OPCDS)     /* OPC Group names        */
PATHIN  DEVICE(cccc,dddd)
PATHOUT DEVICE(aaaa,bbbb)
PATHIN  STRNAME(str1,str2) CLASS(DEFAULT)
PATHOUT STRNAME(str1,str2) CLASS(DEFAULT)
```

Issue the console command "D XCF,CLASSDEF,CLASS=ALL" to see if you already have a DEFAULT class (the name of this class might be something other than DEFAULT) having CLASSLEN(956), which is the default value. If there is such a

class, you just need to add the TWS specific GROUP names (OPCGRP,OPCDS) to the CLASSDEF statement for that CLASS, as shown in the example above.

Note: By specifying MAXMSG(2000) on the COUPLE statement, as shown above, all transport classes will use this value unless a different value is specified at the CLASSDEF level. MAXMSG(2000) is the default value.

If XCF is used to connect the Data Store to the controller, a specific XCF group must be defined, and it must be different from the one used to connect the controller to the z/OS tracker. These two separate XCF groups can use the same XCF transport class.

Note: You can change XCF options while the system is active by using the SETXCF operator command.

For more information about XCF, see *z/OS MVS Setting up a Sysplex*.

Modifying TSO parameters

You must define the EQQMINOM module to TSO on each system where you install the scheduler dialogs. You must also authorize the IBM Workload Scheduler for z/OS TSO commands on every system where you install IBM Workload Scheduler for z/OS. If you do not authorize the TSO commands, the commands will only work on the system where the controller is installed.

To request services from the subsystem for a TSO user, the IBM Workload Scheduler for z/OS dialog invokes the EQQMINOM module using the TSO service facility. EQQMINOM is the dialog interface module. It must run as an APF-authorized program. To achieve this, define EQQMINOM to TSO. If you are installing the scheduler dialogs, include EQQMINOM in the list of programs defined by the AUTHTSF statement in the IKJTSONn member of SYS1.PARMLIB. This statement defines programs to be authorized when invoked using the TSO service facility, as shown in the following example:

```
IKJTSONn AUTHTSF example
AUTHTSF NAMES(IKJEFF76 +
              IEBCOPY +
              EQQMINOM)
```

If you prefer, you can put EQQMINOM in CSECT IKJEFTAP instead of IKJTSONn. For more information about using IKJEFTAP, see *TSO/E Customization*.

IBM Workload Scheduler for z/OS supports the BACKUP, BULKDISC, JSUACT, OPINFO, OPSTAT, SRSTAT, and WSTAT TSO commands. Update the IKJTSONn member on each system where you are installing IBM Workload Scheduler for z/OS to define these commands as authorized commands. To do this, add them to the list of commands defined by the NAMES keyword of the AUTHCMD statement, as shown in the following example

```
IKJTSONn AUTHCMD example
AUTHCMD NAMES(BACKUP +
              BULKDISC +
              JSUACT +
              OPINFO +
              OPSTAT +
              SRSTAT +
              WSTAT)
```

If the default entry in the ISPF TSO command table ISPTCM is set for unauthorized TSO commands, then ISPTCM must be updated. The ISPTCM can be

updated using the ISPMTCM macro. Define the BACKUP, BULKDISC, JSUACT, OPINFO, OPSTAT, SRSTAT, and WSSTAT commands like this:

```
ISPTCM example
ISPMTCM FLAG=62,ENTNAME=BACKUP
ISPMTCM FLAG=62,ENTNAME=BULKDISC
ISPMTCM FLAG=62,ENTNAME=JSUACT
ISPMTCM FLAG=62,ENTNAME=OPINFO
ISPMTCM FLAG=62,ENTNAME=OPSTAT
ISPMTCM FLAG=62,ENTNAME=SRSTAT
ISPMTCM FLAG=62,ENTNAME=WSSTAT
```

No update is needed to ISPTCM if the default entry is set up for authorized TSO commands. For more information about the ISPMTCM macro statements, see *ISPF Planning and Customization*.

Performance considerations

The tracker and the controller address spaces must be nonswappable. To do this, include the definition of their top load module, EQQMAJOR, in the program properties table (PPT). This PPT entry example is defined in a SCHED nn member of SYS1.PARMLIB:

```
SCHEDnn example
PPT PGMNAME(EQQMAJOR) NOSWAP
```

The EQQMAJOR program must run in storage key 8, the default value.

To ensure prompt processing by IBM Workload Scheduler for z/OS and to avoid delays in the handling of event records, the tracker subsystem performance rating (that is, its dispatching priority) should match that of the JES subsystem.

Defining the DLF exit for Hiperbatch support

If you want to include Hiperbatch support for IBM Workload Scheduler for z/OS controlled jobs, specify the DLF exit name in the COFDLF nn member of SYS1.PARMLIB. A DLF exit sample is supplied with the SEQQSAMP library. The exit must reside in an authorized library in the LNKLST concatenation. This example of a COFDLF nn member defines a DLF exit called OPCDLF:

```
COFDLFnn example
CLASS MAXEXPB(nnnn) PCTRETB(nnn) CONEXIT(OPCDLF)
```

For more information about invoking Hiperbatch support in IBM Workload Scheduler for z/OS, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Starting the product automatically

The COMMND nn member of SYS1.PARMLIB list z/OS commands automatically issued during system initialization. To avoid delays in starting IBM Workload Scheduler for z/OS when the z/OS system is started, consider including the names of your IBM Workload Scheduler for z/OS started tasks in this member. For information on how to include start commands for your IBM Workload Scheduler for z/OS address spaces, see *MVS Initialization and Tuning Reference*.

Updating APPC options

If you want to use the API, or the server, to communicate with IBM Workload Scheduler for z/OS, you must update APPC options. See “Step 17. Activating support for the API” on page 147, or “Step 18. Activating support for the product dialog and programming interface using the server” on page 150, for a detailed description of what you need to do.

Implementing support for data set triggering

Use the IBM Workload Scheduler for z/OS data set triggering function to start dependent processing or schedule unplannable work by automatically generating special resource availability events when a data set is closed after being opened for:

- Read processing
- Output processing
- Either read or output processing.

IBM Workload Scheduler for z/OS uses the SMF exit IEFU83 to generate a resource availability event when IEFU83 is called for SMF record types 14, 15, or 64. The data set activity SMF records are generated when a data set is closed or processed by EOVS. IBM Workload Scheduler for z/OS will generate resource availability events only when the data set is closed. When a VSAM data set is closed, two SMF 64 records are created, one each for the DATA and INDEX components. When resource availability events are requested for VSAM data sets, the event will be created when the DATA component is closed, IBM Workload Scheduler for z/OS will not generate an event when the INDEX component is closed.

SMF data set activity records are written when the data set is closed, regardless of whether the JOB/STEP/TASK/USER completed successfully. For more information about the data sets that generate SMF record types 14, 15, or 64, see the documentation for MVS SMF.

To define the data sets for which you want events to be generated, you can perform either of the following:

- Use the EQQRXTRG program to centralize and automate the population of the data set to which the EQQJCLLIB DD name refers. For detailed information about running event-driven workload automation, see *Managing the Workload*.
- Build a selection table, as described in Appendix D, “Invoking the EQQLSENT macro,” on page 391. The selection table is located in ECSA. It is automatically loaded from the data set referred to by the EQQJCLLIB DD name when the event writer is started in a tracker if a table has not previously been loaded since IPL. To reload the table at any time, issue the z/OS modify command:

```
F procname,NEWDSLST
```

Note: No support is available for the data set triggering function before the event writer is started immediately after a z/OS IPL. When the event writer has started after IPL, data set triggering functions are available if the event writer is subsequently stopped. To stop data set triggering at any time issue the NEWDSLST modify command to load a table that contains only the end-of-table indicator.

To implement support for the data set triggering function, perform these actions:

- Update SYS1.PARMLIB member SMFPRMnn as described in “Updating SMF parameters” on page 86.
- Install SMF exit IEFU83 using the EQQU831 sample. See “Macro invocation syntax for EQQEXIT” on page 388 on how to specify the SRREAD parameter.
- Define the data set selection criteria as described by the event-driven resource handling section in *Managing the Workload*.

The procedure described in Appendix D, “Invoking the EQQLSENT macro,” on page 391 is supported for compatibility with earlier versions only.

Step 7. Setting up the RACF environment

If your installation protects data and resources from unauthorized use, you must define IBM Workload Scheduler for z/OS to your security system. This section assumes that the Resource Access Control Facility (RACF) is installed and active on your z/OS system. It describes the activities you must perform to define and enable the security environment for IBM Workload Scheduler for z/OS.

IBM Workload Scheduler for z/OS: Customization and Tuning contains detailed plans and instructions for establishing a security strategy for your IBM Workload Scheduler for z/OS resources.

Controlling the user ID of the address space

If you run IBM Workload Scheduler for z/OS as a started task, you must associate the cataloged procedure name with a suitably authorized RACF user. The user ID must be defined in the STARTED resource class.

If you use any of the following definitions in your initialization statements:

- TPLGYSRV parameter in the OPCOPTS statement
- TCPIP parameter in the ROUTOPTS statement
- MONOPTS statement

you must also define an OMVS segment for the controller user ID.

Controlling the user ID of submitted jobs

IBM Workload Scheduler for z/OS can submit three kinds of jobs:

- Normal production jobs, which are submitted when their prerequisites in the current plan are fulfilled.
- Stand-alone cleanup jobs, which are submitted to run cleanup actions separately from the original job.
- Dialog jobs, which you can submit directly from a panel in the IBM Workload Scheduler for z/OS dialog.

Normal production jobs

IBM Workload Scheduler for z/OS submits production jobs to the internal reader, or starts started tasks, when all prerequisites are fulfilled. The JCL comes from the JS file (EQQJSnDS), the JCL job library (EQQJBLIB), or the job-library-read exit (EQQUX002). You can determine the authority given to a job or started task in several ways:

- You can submit work with the authority of the IBM Workload Scheduler for z/OS address space. The job or started task is given the same authority as the controller or tracker whose submit subtask actually submits the work. For example, work that is transmitted from the controller and then submitted by the tracker is given the authority of the tracker.
- Another method is to use the job submit exit, EQQUX001. This exit is called when IBM Workload Scheduler for z/OS is about to submit work.
 - You can use the RUSER parameter of the EQQUX001 exit to cause the job or started task to be submitted with a specified user ID. The RUSER name is supported even if the job or started task is first sent to a tracker before being started.
 - In certain circumstances you might need to include a password in the JCL to propagate the authority of a particular user. You can use the job-submit exit (EQQUX001) to modify the JCL and include a password. The JCL is saved in

the JCL repository (JSn) data set *before* the exit is called, thus avoiding the need to store JCL with specific passwords. This method prevents the password from being visible externally. For more information about the job-submit exit, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Stand-alone cleanup jobs

Their purpose is to run data set cleanup actions and can be submitted when:

- An automatic internal process takes place (for example, when cleanup type immediate is used and an operation ends in error)
- A Start Cleanup command is issued by an IBM Workload Scheduler for z/OS dialog or the Dynamic Workload Console.

Activate exit EQQUX001 to make sure that the submitter of the stand-alone cleanup job is the same as the submitter of the original job, otherwise the stand-alone cleanup job will run with the same authority as the controller or the tracker that submits it. The current EQQUX001 sample contains a procedure to set the RUSER value according to the value of the USER= keyword in the jobcard of the original job.

Dialog jobs

When you submit IBM Workload Scheduler for z/OS batch jobs from your TSO address space, they go through normal TSO functions. This means that you can submit any job allowed by TSO/E. IBM Workload Scheduler for z/OS makes no authority checks when the job is submitted.

For the IBM Workload Scheduler for z/OS batch job to run successfully, it must be authorized to reference the data sets it uses. The submitting TSO user might also need authorization to use a specific function. For example, a user could have update authority to the AD file but not have the authority to use the AD mass update function.

Protecting data sets

For basic security of IBM Workload Scheduler for z/OS data, you should restrict access to all the product data sets.

Two categories of users need different levels of access to the product data sets:

- Software support people must be able to debug problems and reorganize VSAM files. You might give them alter access to all the product data sets.
- Administrators and operators must be able to use the product dialogs. They need read access to ISPF-related data sets (such as the panel and message libraries), but they do not access the databases (such as the workstation database) directly: these files are accessed by the IBM Workload Scheduler for z/OS subsystem, not by any code in the TSO user's address space. Authority to access the data for a dialog user is given using the authorization functions provided by the product.

The IBM Workload Scheduler for z/OS started task needs:

- Alter access to VSAM data sets
- Read access to input data sets, such as the message library (EQQMLIB) and parameter library (EQQPARM)
- Update access to all other IBM Workload Scheduler for z/OS data sets
- Update access to catalogs and alter access to data sets for all work that IBM Workload Scheduler for z/OS tracks, if you use the Restart and Cleanup function.

Controlling access to resources

Before IBM Workload Scheduler for z/OS performs any request initiated by a user, a security verification check is passed to the system authorization facility (SAF) to ensure that the user is authorized to access all resources needed to run the request. A user can request IBM Workload Scheduler for z/OS services from:

- An ISPF dialog session
- TSO commands
- The program interface (PIF)
- The application programming interface (API)
- Dynamic Workload Console

Any security software that interfaces with SAF also works with IBM Workload Scheduler for z/OS. For this section, the security product is assumed to be RACF.

The z/OS router service calls RACF to perform authority checks. It provides an installation exit that you can use instead of, or in addition to, RACF to perform resource control functions.

Use the IBM Workload Scheduler for z/OS reserved resource class IBMOPC.

The default class for IBM Workload Scheduler for z/OS is OPCCLASS. If you use a different class name, you must specify it in the AUTHDEF statement. Generally, this means specifying CLASS(IBMOPC) in the AUTHDEF statement. If you are running more than one IBM Workload Scheduler for z/OS system, for example a test system and production system, you might want to define more than one RACF class. By using different CLASS parameters in each AUTHDEF statement, you can specify a different authorization scheme for each system.

To control access to IBM Workload Scheduler for z/OS functions, give at least one TSO user-class authority to the resource class. This TSO user can then allow other IBM Workload Scheduler for z/OS users to access resources as needed.

IBM Workload Scheduler for z/OS also uses the APPL resource class. Define the subsystem name as a resource in the APPL class. The easiest way to do this is to have the RACF administrator give class authority to the APPL resource class to one TSO user. This TSO user defines the subsystem name (for example, OPCC) to the APPL resource class by entering:

```
/*Define subsystem resource*/  
RDEFINE APPL OPCC UACC(NONE)
```

See *RACF Command Reference* and *RACF Administrator's Guide* if you are unfamiliar with this process.

When the subsystem name is defined to RACF, you can give other TSO users access to IBM Workload Scheduler for z/OS. For example, to allow the TSO user OPCUGRP to access OPCC with an update access authority by default, enter:

```
/*Permit access to IBM Workload Scheduler for z/OS*/  
PERMIT OPCC ID(OPCUGRP) ACCESS(UPDATE) CLASS(APPL)
```

For remote dialog users and remotely run PIF applications, the server will do the authority checking; it will check both the APPL class subsystem name resource and the scheduler fixed resources. The user for which the server does authority checking is:

- For dialog users, the TSO user ID.

- For PIF applications, the user ID defined in the security environment of the PIF job.

Permitting access to the controller through the API

If you use the API, you can control access to the controller through the security functions of both APPC/MVS and IBM Workload Scheduler for z/OS. Ensure that you consider both these environments when you update RACF. For more information about controlling access to IBM Workload Scheduler for z/OS through the API, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Controlling access to IBM Workload Scheduler for z/OS resources when using the Dynamic Workload Console

The WebSphere® Application Server performs a security check when a user tries to use Dynamic Workload Console, checking the user ID and password. The WebSphere Application Server associates each user ID and password to an administrator.

The scheduler resources are currently protected by RACF.

The Dynamic Workload Console user should only have to enter a single user ID and password combination, and not provide two levels of security checking (at the WebSphere Application Server level and then again at the IBM Workload Scheduler for z/OS level).

The security model is based on having the WebSphere Application Server security handle the initial user verification, while at the same time obtaining a valid corresponding RACF user ID. This makes it possible for the user to work with the security environment in z/OS.

z/OS security is based on a table mapping the administrator to a RACF user ID. When a WebSphere Application Server user tries to initiate an action on z/OS, the administrator ID is used as a key to obtain the corresponding RACF user ID.

The server uses the RACF user ID to build the RACF environment to access IBM Workload Scheduler for z/OS services, so the administrator must relate, or map, to a corresponding RACF user ID.

For information on how to get the RACF user ID, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Permitting access to the controller through the Dynamic Workload Console

If you use the Dynamic Workload Console, you can control access to the controller through the security functions of both WebSphere Application Server and IBM Workload Scheduler for z/OS. Ensure that you consider both these environments when you update RACF. For more information about controlling access to IBM Workload Scheduler for z/OS through the Dynamic Workload Console, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Authorizing IBM Workload Scheduler for z/OS as a job submitter

Consider the following resource classes when implementing security for IBM Workload Scheduler for z/OS. The examples assume that the RACF user for the IBM Workload Scheduler for z/OS address space is OPCAPPL, which is the name specified in the started-procedure table.

JESJOBS

If your installation has activated the JESJOBS class, you must permit IBM Workload Scheduler for z/OS to submit all jobs that are defined in the current plan. One way of doing this is to permit IBM Workload Scheduler for z/OS to submit all jobs. You can do this by:

1. Defining the submit resource:

```
RDEFINE JESJOBS SUBMIT.*.* UACC(NONE) OWNER(OPCAPPL)
```

2. Authorizing IBM Workload Scheduler for z/OS:

```
PERMIT SUBMIT.*.* CLASS(JESJOBS) ID(OPCAPPL) ACC(READ)
```

SURROGAT

A *surrogate job submission* occurs when all the following conditions are met:

1. USER=xxxx is specified on the job card of the submitted job.
2. The xxxx is not the same as the submitting (RACF) user.
3. No password is specified on the job card.

You might use the job-submit exit (EQQUX001) to return a submitting user in the RUSER field. This is required if you want stand-alone cleanup jobs to be submitted with the same authority as the original job, otherwise you can replace it with surrogate job submission.

To permit IBM Workload Scheduler for z/OS to submit this job, perform the following steps:

1. Activate the surrogate class:

```
SETROPTS CLASSACT(SURROGAT)
```

2. Define the submit resource:

```
RDEFINE SURROGAT APLUSER.SUBMIT UACC(NONE) OWNER(APLUSER)
```

3. Authorize IBM Workload Scheduler for z/OS:

```
PERMIT APLUSER.SUBMIT CLASS(SURROGAT) ID(OPCAPPL) ACC(READ)
```

If the PRIVILEGED or TRUSTED attribute is set in the Started Procedure Table (SPT) entry, the IBM Workload Scheduler for z/OS is authorized to submit jobs under any user regardless of what is defined in the resource rules.

For further information, see the *RACF Administrator's Guide*.

Authorizing IBM Workload Scheduler for z/OS to issue JES commands

Consider the following resource classes when implementing security for IBM Workload Scheduler for z/OS. The examples assume that the RACF user for the IBM Workload Scheduler for z/OS address space is OPCAPPL, which is the name specified in the started-procedure table.

OPERCMDS

If the OPERCMDS class is active and you have specified HOLDJOB(YES) or HOLDJOB(USER) for an event writer, the IBM Workload Scheduler for z/OS address space where the event writer is started must be authorized to issue the JES release command. One method is to permit IBM Workload

Scheduler for z/OS to issue all JES commands. To permit IBM Workload Scheduler for z/OS to issue JES commands on a JES2 system, perform the following steps:

1. Define the resource:

```
RDEFINE OPERCMDS JES2.* UACC(NONE)
```
2. Authorize IBM Workload Scheduler for z/OS:

```
PERMIT JES2.* CLASS(OPERCMDS) ID(OPCAPPL) ACC(UPDATE)
```

On a JES3 system, replace JES2.* with JES3.* in the example. Alternatively, you could specify the JES%.* resource name for either a JES2 or JES3 system.

If you use IBM Workload Scheduler for z/OS to schedule started tasks, the address space must be authorized to issue the z/OS start command. One way of doing this is to permit IBM Workload Scheduler for z/OS to issue all z/OS commands. To do this, perform the following steps:

1. Define the resource:

```
RDEFINE OPERCMDS MVS.* UACC(NONE)
```
2. Authorize IBM Workload Scheduler for z/OS:

```
PERMIT MVS.* CLASS(OPERCMDS) ID(OPCAPPL) ACC(UPDATE)
```

Authority to use the z/OS start command is also required if you use Hiperbatch support for IBM Workload Scheduler for z/OS operations.

JESSPOOL

If the JESSPOOL class is active and you use the IBM Workload Scheduler for z/OS JCC function, you must authorize IBM Workload Scheduler for z/OS to access SYSOUT data sets for all jobs in the current plan. One way of doing this is to permit IBM Workload Scheduler for z/OS to access all SYSOUT data sets. To permit IBM Workload Scheduler for z/OS to access all SYSOUT data sets, perform these steps on each system where the JCC is started:

1. Define the resource:

```
RDEFINE JESSPOOL *.* UACC(NONE)
```
2. Authorize IBM Workload Scheduler for z/OS:

```
PERMIT *.* CLASS(JESSPOOL) ID(OPCAPPL) ACC(ALTER)
```

If the PRIVILEGED or TRUSTED attribute is set in the Started Procedure Table (SPT) entry for IBM Workload Scheduler for z/OS, then the address space is authorized to issue any commands and to process spool data sets regardless of what is defined in the resource rules.

For further information, see the *RACF Security Administrator's Quick Reference*.

Authorizing IBM Workload Scheduler for z/OS E2E server task to create USS processes

In a RACF environment you can define profiles in the UNIXPRIV class to grant RACF authorization for certain z/OS UNIX privileges. If the UNIXPRIV class is active, the user ID of the E2E server task (eqqUID, as specified in the EQQPCS05 job) must have at least READ authorization for the SUPERUSER.FILESYS and SUPERUSER.PROCESS.* profiles, otherwise the user ID cannot create the USS processes.

Make sure that you use a unique UID with a nonzero value; for additional information about this requirement, see INFO APAR II14235.

Authorizing IBM Workload Scheduler for z/OS E2E and Dynamic Workload Console server tasks for security resource EZB.BINDDVIPARANGE

You must give UPDATE authorization for the EZB.BINDDVIPARANGE resource to the user ID of the end-to-end server when using DVIPA host names. Specifically, this authorization is always needed when the TOPOLOGY HOSTNAME value represents a DVIPA address.

If you use the Dynamic Workload Console, you must give UPDATE authorization for the EZB.BINDDVIPARANGE to the user ID of the Dynamic Workload Console server when using DVIPA hostnames. Specifically, this authorization is always needed when the SERVOPTS JSCHOSTNAME value represents a DVIPA address.

Step 8. Securing communications

IBM Workload Scheduler for z/OS supports authentication and cryptography by activating the Secure Sockets Layer (SSL) transport protocol for transmitting and accepting secure information.

You can configure IBM Workload Scheduler for z/OS to enable SSL communication in a TCP/IP network or, you can implement SSL security for HTTP connections as required.

Security for TCP/IP connections

The scheduler authentication mechanism uses the SSL services of z/OS. For further details, see *z/OS Cryptographic Services System Secure Sockets Layer Programming*.

To enable SSL authentication for your network, perform the following actions:

1. Create the SSL work directory by using the EQQPCS10 sample JCL. You can use the same directory as the one used for SSL in E2E. In the following examples, the directory is: /u/tws/ssl
2. From /u/tws/ssl/ as current directory, open a shell prompt, start the **gskkyman** utility of z/OS Cryptographic Services System SSL, and do the following:
 - a. Create the keystore database and consider protecting it from unauthorized access, because it has to contain private key and trusted certificates. For example, consider the following database: /u/tws/ssl/TWS1.kdb.
 - b. Generate a password file and store it in the SSL directory defined in the previous step, for example /u/tws/ssl/TWS1.sth.
 - c. At this point you can:
 - Create a certificate request and send it to the Certificate Authority.
 - Store the signed certificate in the database.
 - Import the certificate of the Certification Authority which signed your certificate.

In this way you have a database containing both your certificate and Certification Authority's one.

The scheduler uses a default name to identify your certificate; therefore you are not required to set up a multiple database handling. If you need different certificates in order to partition your network from a security point of view, you need different databases. The advantage of this solution is that you can store each database in a different directory, with its own access list.

3. Configure IBM Workload Scheduler for z/OS, by specifying the TCPOPTS statement for each component of your network. Consider each component according a client-server model. Typically, a client-server group is composed by the trackers and data stores communicating with the corresponding controller, or by a remote interface communicating with the corresponding server.

When the controller or the server started task communicates with a partner component, the communication is always started by the partner component; therefore the partner acts as a client. Differently from the end-to-end communication, the communicating partners use the same port numbers for both non-SSL and SSL communications.

Specify the same TCPOPTS parameters for all the components in a client-server group.

For a detailed description of the TCPOPTS statement, see *Customization and Tuning*. The following example shows a TCPOPTS definition to activate the SSL support.

```
TCPOPTS
. . .
SSLLEVEL(FORCE)
SSLKEYSTORE('/u/tws/ssl/TWS1.kdb')
SSLKEYSTOREPSW('/u/tws/ssl/TWS1.sth')
SSLAUTHSTRING('OPCMaster')
SSLAUTHMODE(String)
```

1
2
3
4
5

In this example:

- 1 The FORCE keyword enables the SSL communication.
- 2 TWS1.kdb is the database containing the certificate.
- 3 TWS1.sth is the password file to access the database.
- 4 OPCMASTER is the string defined as Common Name (CN) in the certificate.
- 5 The STRING keyword enables the check on the CN string.

When designing your configuration from a security point of view, consider that:

- To enforce your security, you can use the SSLAUTHMODE(String), that requires to:
 - Create an SSL certificate for each controller started task. This certificate will be used by the controller and its remote partners. Define the certificate using as Common Name a unique string corresponding to the controller.
 - Create an SSL certificate for each server started task. This certificate will be used by the server and its remote partners. Define the certificate using as Common Name a unique string corresponding to the server.

The SSLAUTHSTRING must match the information contained in the certificate sent by the partner. To verify it, you can use the **gskkyman** utility that allows displaying the keys database and SSL certificate content. The certificate CN is returned by **gskkyman** as the first line of the “Subject”.

- If you prefer to use SSLAUTHMODE(CAONLY), then you can use a single SSL certificate for all your network.

Security for HTTP connections

You can provide security for an HTTP connection between the following components:

- The z/OS controller and the IBM Workload Scheduler for z/OS Agent.
- The z/OS controller and another z/OS controller (z/OS remote engine).
- The z/OS controller and the dynamic domain manager.
- The z/OS controller and the IBM Workload Scheduler master domain manager (distributed remote engine).

SSL-secure connections are implemented using specific settings in the HTTPOPTS initialization statement, and the HTTPS keyword in the ROUTOPTS initialization. For more information about these statements, see *Customization and Tuning*.

If you use the secure connection with the SSL protocol, you must import the security certificates into your security system.

Note: If you imported the default security certificates during the installation of the previous version of the product, you must remove them and run the EQQRCERT job to import the new certificates. If you already imported the new default security certificates during the installation of the IBM Workload Scheduler agent for z/OS, then you must not perform this procedure again. Complete the procedure for creating a secure connection by configuring the SSLKEYRING keyword with the value used for installation of the IBM Workload Scheduler agent for z/OS.

At installation time, the default security certificates are automatically stored into the SEQQDATA library:

EQQCERCL

The security certificate for the client.

EQQCERSR

The security certificate for the sever.

You can decide to use these default certificates or create your own. In both cases, you must import them into your security system. If you are using RACF, you are provided with the sample job EQQRCERT to import the certificates. To run this job, ensure that you use the same user ID that RACF associates with the controller started task.

If you create your own certificates for an HTTP connection with the master domain manager or with the dynamic domain manager, you must run the customizing steps described in the section about customizing SSL connection to the master domain manager and dynamic domain manager in *IBM Workload Scheduler: Administration Guide*.

If you are using SSL to communicate with a master domain manager, backup master domain manager, or dynamic domain manager, then the prefix of the common name of the controller certificate must be defined in the **Broker.AuthorizedCNs** option in the BrokerWorkstation.properties file located in the *TWA_home*/TDWB/config directory of the distributed engine.

The EQQRCERT job performs the following actions:

- Copies the EQQCERCL certificate to a temporary sequential data set
- Copies the EQQCERSR certificate to a temporary sequential data set

- Imports EQQCERCL to RACF
- Imports EQQCERSR to RACF
- Deletes the temporary sequential data sets
- Creates the SAF key ring that is used to connect the imported certificates
- Updates the RACF database with the new certificates and key ring

Step 9. Allocating data sets

Note: A standby controller uses the same data sets as the controller.

At this stage of the installation of your IBM Workload Scheduler for z/OS system, you allocate the data sets that your JCL procedures refer to. You can create the data sets by using the jobs created by the EQQJOBS installation aid.

If you are using the EQQJOBS installation aid, you will already have generated several members in the output library that you specified.

Consider carefully where IBM Workload Scheduler for z/OS data sets are allocated in your production environment. Some data sets can be highly active. Avoid placing these data sets on DASD volumes with high activity because this can result in poor performance due to contention. Also consider the ability to recover data sets if a DASD volume becomes unusable. If you place all your data sets on the same volume, you must recover many data sets before you can continue your IBM Workload Scheduler for z/OS service. *IBM Workload Scheduler for z/OS: Customization and Tuning* describes recovery of IBM Workload Scheduler for z/OS data sets in detail.

The space to allocate for your data sets depends upon the workload at your installation. It is difficult to give precise figures for the amount of space you will need. The space allocated by the sample JCL should give you enough space to at least get started. These amounts will be enough for the IBM Workload Scheduler for z/OS service for many installations. Use Table 26 on page 105 as a guide to allocate space for VSAM data sets.

The following sections describe the IBM Workload Scheduler for z/OS data sets and include examples of the JCL needed to create them.

Allocating the VSAM data sets

Perform this task if you are installing a controller.

Table 25 shows the VSAM data sets and their characteristics. The JCL procedure for the controller uses all of these data sets except for EQQLDDS and EQQLTBKP, which are used only in the planning batch jobs. Allocate all these VSAM data sets for a controller.

Table 25. IBM Workload Scheduler for z/OS VSAM data sets

Sample	DD name	Record type	Attributes	Share option	Keys	Record size	Data set
EQQPCS09	N/A	KSDS	REUSE NSPND	3	19 0	200 32000	Archive of current plan
EQQPCS01	EQQADDS	KSDS	UNIQUE SPANNED	3	25 0	1000 131072*	Application description

Table 25. IBM Workload Scheduler for z/OS VSAM data sets (continued)

Sample	DD name	Record type	Attributes	Share option	Keys	Record size	Data set
EQQPCS01	EQQCP1DS	KSDS	REUSE NSPND	3	19 0	200 32000	Current plan 1
EQQPCS01	EQQCP2DS	KSDS	REUSE NSPND	3	19 0	200 32000	Current plan 2
EQQPCS01	EQQCXDS	KSDS	REUSE NSPND	3	64 0	500 32000	Current plan extension
EQQPCS01	EQQXD1DS	KSDS	REUSE NSPND	3	68 0	500 32000	Extended data 1
EQQPCS01	EQQXD2DS	KSDS	REUSE NSPND	3	68 0	500 32000	Extended data 2
EQQPCS01	EQQNXDDS	KSDS	REUSE NSPND	3	68 0	500 32000	New extended data
EQQPCS01	EQQJS1DS	KSDS	REUSE SPANNED	3	28 0	804 180004	JCL repository 1
EQQPCS01	EQQJS2DS	KSDS	REUSE SPANNED	3	28 0	804 180004	JCL repository 2
EQQPCS01	EQQLDDS	KSDS	REUSE SPANNED	2	28 0	440 131072	Long-term-plan work
EQQPCS01	EQQLTBKP	KSDS	REUSE SPANNED	3	28 0	200 131072	Long-term-plan backup
EQQPCS01	EQQLTDS	KSDS	REUSE SPANNED	3	28 0	200 131072	Long-term plan
EQQPCS01	EQQNCPDS	KSDS	REUSE NSPND	3	19 0	200 32000	New current plan
EQQPCS01	EQQNCXDS	KSDS	REUSE NSPND	3	64 0	500 32000	New current plan extension
EQQPCS01	EQQNSTDS	KSDS	UNIQUE SPANNED	3	68 0	500 32000	New step awareness
EQQPCS01	EQQOIDS	KSDS	UNIQUE NSPND	3	28 0	800 32000	Operator instruction
EQQPCS07	EQQPKIxx	KSDS	UNIQUE INDEXED	1,3	34 0	77 77	Primary Index
EQQPCS01	EQQRDDS	KSDS	UNIQUE NSPND	3	64 0	400 32000	Special resource descriptions
EQQPCS01	EQQSCPDS	KSDS	REUSE NSPND	3	19 0	200 32000	Current plan backup copy for Symphony creation and for IBM Tivoli Monitoring integration
EQQPCS07	EQQSDFxx	LINEAR	N/A	2,3	N/A	N/A	Data files
EQQPCS01	EQQSIDS	KSDS	UNIQUE NSPND	3	64 0	110 220	Side information file: ETT and configuration information
EQQPCS07	EQQSKIxx	KSDS	UNIQUE INDEXED	1,3	38 0	76 32000	Secondary Index

Table 25. IBM Workload Scheduler for z/OS VSAM data sets (continued)

Sample	DD name	Record type	Attributes	Share option	Keys	Record size	Data set
EQQPCS01	EQQSTDS	KSDS	UNIQUE SPANNED	3	68 0	500 32000	Step awareness
EQQPCS01	EQQWSDS	KSDS	UNIQUE NSPND	3	10 0	100 32000	Workstation, calendar, and period descriptions.
EQQPCS01	EQQSCPDS	KSDS	REUSE NSPND	3	19 0	200 32000	Current plan backup copy for Symphony creation and for IBM Tivoli Monitoring integration

Note:

- * The maximum record size for EQQPCS01 is the default maximum value. This can be increased as in the example that follows.
- In specific situations where the size of the CP files (CP1, CP2, NCP, SCP) are large and the batch daily planning jobs cause a considerable number of updates to the NCP, it is possible for the NCP to become very large. This might require the allocation of additional extents (not additional volumes, since ADDVOL support is not available for the NCP file). Consider freespace allocation for the current plan, including NCP (EQQCP1DS, EQQCP2DS, EQQCXDS, EQQNCPDS and EQQSCPDS), application descriptions (EQQADDS), resource descriptions (EQQRDDS), and operator instructions (EQQOIDS) data sets.
- The extended data sets XD1DS, XD2DS, NXDDS, OXDDS have a logical correspondence and use like the current plan data sets CP1DS, CP2DS, NCPDS, ONCPDS. As the old CP (OCP) can be either the CP1 or the CP2 according to which one is inactive and not current, with the same logic OXD can be either XD1 or XD2 according to which one is not currently active.
- If you are upgrading from a previous IBM Workload Scheduler for z/OS release, it is recommended to use the new samples shipped and set with EQQJOBS. The allocation samples like EQQPCS01 use variables. If you are customizing previous allocation JCLs, make sure you correctly position the changes and use the correct set of defined variables.

You can allocate the VSAM data sets by submitting the sample listed in Table 25 on page 102. Alternatively, you can allocate one or more of the VSAM data sets by running a job like this:

```

Allocating a VSAM data set
//ALOCVSAM JOB STATEMENT PARAMETERS
/*-----*
/* ALLOCATE AN OPC VSAM DATA SET *
/*-----*
//ALLOC EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=Q
//EQQVOL1 DD DISP=OLD,VOL=SER=volser,UNIT=3390
//SYSIN DD *
DEFINE +
  CLUSTER ( +
    NAME('OPC.INST.AD') UNIQUE +
    SPANNED +
    SHR(3) VOL(volser) CYLINDERS(2 2) +
  ) +
  DATA ( +
    NAME('OPC.INST.ADDATA') +
    KEYS(25 0) RECORDSIZE(1000 132072) +
  ) +
  INDEX ( +
    NAME('OPC.INST.ADINDEX') +
  )
/*

```

This example allocates the application description database.

You can allocate VSAM data sets on different device types.

Allocate enough space for your data sets, depending upon the amount of work IBM Workload Scheduler for z/OS processes at your installation. You can use Table 26 as a guide to allocate space for VSAM data sets.

Table 26. Calculations of VSAM data set size

Data set	Size in bytes is total of:	
	Number of	Multiplied by
Application description (EQQADDS)	Application and group definitions	208
	Run cycles	120
	Positive run days	3
	Negative run days	3
	Operations	110
	Internal dependencies	16
	External dependencies	84
	Special resources	64
	Operation Extended Information	200
	Variable tables	98
	Variables	476
	Variable dependencies	88
	Extended Name	
Current plan (EQQCPnDS)	Header record (one only)	188
	Workstations	212
	Workstation open intervals	48
	Workstation access method data	72
	Occurrences	302
	Operations	356
	Dependencies	14
	Special resource references	64
	Operation Extended Information	200
	Jobs	116
	Executed steps	20
	Print operations	20
	Unique application names	64
	Operations currently in error	264
	Reruns of an operation	264
	Potential predecessor occurrences	32
	Potential successor occurrences	24
	Operations for which job log information has been collected	111
Stand-alone clean up	70	
Restart and clean up operinfo retrieved	44	
Number of occurrences	43	
Extended data (EQQXDnDS and EQQNXDDS)	Header record (one only)	244
	Bind requests	565
JCL repository (EQQJSnDS)	Number of jobs and started tasks	80
	Total lines of JCL	80
	Operations for which job log information has been collected	107
	Total lines of job log information	143
Note: As a base, calculate a figure for all your jobs and started tasks controlled by IBM Workload Scheduler for z/OS. Add to this figure the expected space required for jobs and started tasks in the current plan.		

Table 26. Calculations of VSAM data set size (continued)

Data set	Size in bytes is total of:	
	Number of	Multiplied by
Long-term plan (EQQLTDS)	Header record (one only)	92
	Occurrences	160
	External dependencies	35
	Operations changed in the LTP dialog	58
Operator instruction (EQQOIDS)	Instructions	78
	Instruction lines	72
Special resource database (EQQRDDS)	Resource definitions	216
	Defined intervals	48
	Entries in the WS connect table	8
Side information file (EQQSIDS)	ETT requests	128
Workstation/calendar (EQQWSDS)	Calendars	96
	Calendar dates	52
	Periods	94
	Period origin dates	6
	Workstation closed dates	80
	Workstations	124
	Workstation access method data	72
	Interval dates	52
	Intervals	32
Note:		
1. Use the current plan data set calculation (EQQCPnDS) for the new current plan data sets (EQQNCPDS and EQQSCPDS).		
2. Use the long-term-plan data set calculation (EQQLTDS) for the long-term-plan work data set (EQQLDDS) and the long-term-plan backup (EQQLTBKP).		
3. Use the special resource database calculation (EQQRDDS) for the current plan extension data set (EQQCXDS) and the new current plan extension (EQQNCXDS).		

Consider the following information when allocating VSAM data sets.

Archive of current plan (EQQACPDS)

The ACP file is a copy of the old current plan that is created by the database archive job (EQQDBARC). The file name is passed to the job with the parameter VSAMBKP.

Application description data set (EQQADDS)

The application description data set contains application descriptions and JCL variable tables. This data set is allocated as a spanned data set by EQQPCS01. It has a default maximum record size of 131 072. This allocation limits the variable definitions in a variable table to 275 (131 072/476 = 275), provided there are no variable dependencies. If you also use variable dependencies, the number of variables in a JCL variable table is less than 275.

If you will use a greater number of variable definitions in a variable table, allocate the application description data set with a greater record size. To calculate how great the record size should be, use this method:

$$\text{LRECL} = 86 + (\text{maximum number of variables in one table} * 476) +$$

$$(\text{number of variable dependencies} * 88)$$

where 86 is the length of the header record, 476 is the length of each variable record and 88 is the length of each variable dependency record.

This VSAM data set must be allocated with share option set to 3 SHR(3). Do not use share option 2 or 1.

Current plan data sets (EQQCP n DS)

The current plan VSAM files are opened and closed many times by IBM Workload Scheduler for z/OS during normal processing. If IBM Workload Scheduler for z/OS is unable to open one of the files, for example if the file is already opened by another job or TSO user, the normal mode manager (NMM), is terminated. The NMM issues message EQQN027E which reports the reason for the unexpected termination. You can issue a MODIFY command to restart the NMM subtask.

It is recommended that you do not access the current plan files from outside the IBM Workload Scheduler for z/OS address space. Backups of current plan information should be taken from the new current plan (EQQNCPDS). Shut down the controller address space if full-pack backups are taken of the volumes where the data sets reside.

Note: An extended-format data set for VSAM can be allocated for CP n data sets that exceed 4 GB.

JCL repository data sets (EQQJS n DS)

Take special care when allocating the JCL repository data sets. The following information describes the allocation and use of these data sets.

IBM Workload Scheduler for z/OS maintains its own copy of JCL in the JCL repository data set for every job that it submits in the current plan. It uses a primary and alternate data set for the JCL repository, EQQJS1DS and EQQJS2DS. It reorganizes the JCL repository data set that is in use by copying it to the alternate data set and then switching over to use the newly copied data set. The value you specify on the MAXJSFILE keyword defines whether the JCL repository should be automatically copied and determines how often the automatic copy process should occur.

Use the EQQPCS01 sample job created by the EQQJOBS installation aid to allocate the JS data sets. This job allocates the JS data sets with the SPANNED attribute and maximum record size 180 000. This limits the maximum number of JCL statements to 2 249 for any one job. If you run jobs with a greater number of JCL statements, increase the record size. Calculate the required record size, in bytes, by multiplying the number of JCL statements in your largest job by 80, and add an extra 80 bytes for the header record. If you define your JS file without SPANNED, the greatest maximum record size that you can specify is 32 760 bytes. This lets you store a job with up to 408 JCL records. If you define the JS data sets with SPANNED, the maximum record size you can specify is slightly less than a control area (CA). If you use the EQQUX002 exit, the largest job that can be returned by this exit is 7 599 JCL records. Consider this, when you define the maximum record size of the JS data sets.

Note:

An extended-format data set for VSAM can be allocated for JS data sets that exceed 4 GB.

Operator Instruction data set (EQQOIDS)

The operator instruction (OI) database contains operator instructions, each of which corresponds to an operation in the AD database and provides specific instructions about how this operation has to be handled.

This VSAM data set must be allocated with share option set to 3 SHR(3). Do not use share option 2 or 1.

Current plan backup copy data set (EQQSCPDS)

During the creation of the current plan, the SCP data set is used as a CP backup copy for the production of the Symphony file and for the integration with IBM Tivoli Monitoring.

This VSAM data set must be allocated with share option set to 3 SHR(3). Do not use share option 2 or 1.

Note: An extended-format data set for VSAM can be allocated for SCP data sets that exceed 4 GB.

Data sets for step awareness (EQQSTDS and EQQNSTDS)

The EQQSTDS and EQQNSTDS data sets are used to enable the step awareness function. They are updated only by the controllers and appropriate cleanup actions are performed at the end of the CP turnover process. For details about the current plan turnover, see the *Diagnosis Guide and Reference*.

Data sets for extended data (EQQXDnDS)

The extended data VSAM files are opened and closed by IBM Workload Scheduler for z/OS together with the current plan VSAM files. For this reason, the same considerations for the current plan data sets apply to the data sets for the extended data.

Allocating Restart and Cleanup VSAM data sets

Use the EQQPCS07 member generated by the EQQJOBS installation aid. It is contained in the output library specified on the CREATE SAMPLE JOB JCL panel (EQQJOBS8). Submit the EQQPCS07 job to define and initialize the Restart and Clean Up VSAM files.

Note: You can omit this step if you are migrating from a previous IBM Workload Scheduler for z/OS version.

Restart and cleanup data sets (EQQPKlxx, EQQSKlxx, and EQQSDFxx)

Every IBM Workload Scheduler for z/OS address space that uses the Restart and Cleanup function requires the allocation of a local VSAM repository for the structured information related to each job run.

These data sets have the same structure as the Data Store VSAM files and can be allocated by running the EQQPCS07 sample. Keep in mind that every IBM Workload Scheduler for z/OS requires the allocation of a unique local VSAM repository.

The set of data sets allocated by EQQPCS07 is used by the controller started tasks and it is different from the similar set allocated for the data store started task.

Allocating non-VSAM data sets

This section describes the physical sequential (PS) and partitioned (PDS) data sets. Table 27 shows the non-VSAM data sets and their characteristics. Before you allocate the non-VSAM data sets, review the following sections, which contain important information about each of these data sets.

For all the sequential data sets listed below, the current version of IBM Workload Scheduler for z/OS supports DSTYPE LARGE, which allows allocation of sequential data sets larger than 65535 tracks.

Table 27. IBM Workload Scheduler for z/OS non-VSAM data sets

Sample	DD Name	RECFM	LRECL	BLKSIZE	DSORG	Data set	
EQQPCS02	AUDITPRT	FBA	133	13300	PS	Input to EQQAUDIT	
EQQPCS01	–	U	–	6300	PS	CLIST library (optional)	
EQQPCS01	EQQCKPT	U	–	8200	PS	Checkpoint	
EQQPCS01	EQQBKPT	U	–	8200	PS	Backup checkpoint	
	EQQDLnn	U	–	6300	PS	Dual job-tracking-log	
EQQPCS01	EQQDMSG	VBA	84	3120	PS	IBM Workload Scheduler for z/OS diagnostic message and trace	
EQQPCS02	EQQDUMP	FB	80	3120	PS	IBM Workload Scheduler for z/OS diagnostic	
EQQPCS11	EQQDUMP	FB	80	3120	PS	Diagnostic for Output collector	
EQQPCS02	EQQEVDS/ EQQEVDM/ EQQHTTP0	F	100	100	PSU	Event	
EQQPCS01	EQQEVLIB	FB	80	3120	PDS	Event-driven workload automation (EDWA) configuration file repository	
EQQPCS02	EQQINCWK	FB	80	3120	PS	JCC incident work	
EQQPCS01	EQQJBLIB	FB	80	3120	PDS	Job library	
EQQPCS01	EQQJCLIB	FB	80	3120	PDS	JCC message table	
EQQPCS01	EQQJTABL	F	240	240	PS	Critical job table log file	
EQQPCS01	EQQJTARC	U	–	6300	PS	Job-tracking archive	
EQQPCS01	EQQJTnn	U	–	6300	PS	Job-tracking-log	
I	EQQPCS14	EQQDBARC	U	–	6300	PS	Extended-auditing archive
I	EQQPCS14	EQQDBnn	U	–	6300	PS	Extended-auditing log
EQQPCS01	EQQLOGRC	F	128	128	PS	Joblog and Restart Information pending requests Log data set	
EQQPCS02	EQQLOOP	VBA	125	1632	PS	Loop analysis message log	
EQQPCS02	EQQMLOG	VBA	125	1632	PS	Message log	
EQQPCS11	EQQMLOG	VBA	125	1632	PS	Message log for Output collector started task	
EQQPCS01	EQQMONDS	F	160	160	PSU	Monitoring task data set used to store events for IBM Tivoli Monitoring	

Table 27. IBM Workload Scheduler for z/OS non-VSAM data sets (continued)

Sample	DD Name	RECFM	LRECL	BLKSIZE	DSORG	Data set
EQQPCS09	EQQOCPBK	-	-	-	-	Data set to allocate the GDG root. The GDG entry is allocated during DP batch run and contains a backup of the old current plan.
EQQPCS11	EQQOUCFV	F	160	160	PSU	Stores events used in the communication between the controller and Output collector for retrieving job logs from the z-centric environment.
EQQPCS11	EQQOUCKP	FB	80	3120	PDSE	Request checkpoint data set used by Output collector as it reads and processes events in the EQQOUCFV data set.
EQQPCS01	EQQPARM	FB	80	3120	PDS	Initialization-statement library
EQQPCS01	EQQPRLIB	FB	80	3120	PDS	Automatic-recovery-procedure library
EQQPCS06	EQQSCLIB	FB	80	3120	PDS	Script library for end-to-end scheduling with fault tolerance capabilities
EQQPCS01	EQQSTC	FB	80	3120	PDS	Started-task submit
EQQPCS01	EQQSUDS/ <i>user-defined</i>	F	820	820	PSU	Submit/release
EQQPCS02	EQQTROUT	VB	32756	32760	PS	Input to EQQAUDIT
EQQPCS06	EQQTWSCS	FB	80	3120	PDSE	Data set for centralized script support in end-to-end with fault tolerance capabilities
EQQPCS06	EQQTWSIN and EQQTWSOU	F	160, 160	160, 160	PSU	Event data sets for end-to-end with fault tolerance capabilities
-	EQQYPARM				PDS/PS	PIF
EQQPCS01 EQQPCS02	SYSMDUMP	F	4160	4160	PS	System dump data set
EQQPCS11	SYSMDUMP	F	4160	4160	PS	System dump data set for Output collector
-	-	FB	80	3120	PS	Job-completion-checker incident log

You can allocate these non-VSAM data sets using the samples listed in Table 27 on page 109 that are generated by the EQQJOBS installation aid.

Note: The data sets cannot be defined as compressed SMS data sets. If you have not tailored the members as described in “Step 9. Allocating data sets” on page 102, you can allocate a partitioned data set by running a job like this:

```

Allocating an IBM Workload Scheduler for z/OS partitioned data set
//ALLOCPDS JOB STATEMENT PARAMETERS
//*-----*
//* ALLOCATE A PARTITIONED DATA SET *
//*-----*
//ALLOC EXEC PGM=IEFBR14

```

```
//SYSUT1 DD DSN=OPCESA.INST.EQQSTC,
//          DISP=(,CATLG),
//          VOL=SER=volser,
//          SPACE=(TRK,(5,0,1)),
//          UNIT=3390,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

This example allocates a started-task-submit data set (EQQSTC).

To allocate an IBM Workload Scheduler for z/OS sequential data set, you can run a job like the this:

```
Allocating an IBM Workload Scheduler for z/OS sequential data set
//ALLOCPJOB JOB STATEMENT PARAMETERS
/*-----*
/* ALLOCATE A SEQUENTIAL DATA SET *
/*-----*
//ALLOC EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSUT1 DD DUMMY,DCB=(RECFM=F,BLKSIZE=100,LRECL=100)
//SYSUT2 DD DSN=OPCESA.INST.EVENTS,
//          DISP=(NEW,CATLG),
//          UNIT=3390,
//          VOL=SER=volser,
//          SPACE=(CYL,3,,CONTIG),
//          DCB=(RECFM=F,BLKSIZE=100,LRECL=100,DSORG=PS)
//SYSIN DD DUMMY
```

This example allocates an event data set (EQQEVDS). The IEBGENER utility ensures that the allocated data set has an end-of-file marker in it.

Note: If you allocate IBM Workload Scheduler for z/OS data sets using your own jobs, ensure that they have an end-of-file marker in them.

To allocate an IBM Workload Scheduler for z/OS partitioned extended data set, you can run a job such as the following one:

```
Allocating an extended partitioned data set
//ALLOPDSE JOB STATEMENT PARAMETERS
/*-----*
/*ALLOCATE A PDSE DATA SET *
/*-----*
//ALLOC EXEC PGM=IEBR14
//SYSUT1 DD DSN=OPCESA.INST.CS,
// DSNTYPE=LIBRARY,
// DISP=(NEW,CATLG),
// UNIT=3390,
// VOL=SER=volser,
// SPACE=(CYL,(1,1,10)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

This example allocates a data set for centralized script support (EQQTWSCS) in an end-to-end with fault tolerance capabilities environment.

The following sections describe the IBM Workload Scheduler for z/OS non-VSAM data sets. They contain important information to consider when allocating your data sets.

Internal reader data set (EQQBRDS)

When an IBM Workload Scheduler for z/OS subsystem is used to submit work, specify the internal reader data set, EQQBRDS, in your started-task procedures. The DD statement must contain the external-writer-data set name, INTRDR, and

the class of the internal reader. The class you specify is used as a default message class for jobs that do not have a MSGCLASS parameter specified on their job cards.

Example internal reader DD statement
//EQQBRDS DD SYSOUT=(A,INTRDR)

Checkpoint data set (EQQCKPT)

IBM Workload Scheduler for z/OS uses the checkpoint data set to save the current status of the IBM Workload Scheduler for z/OS system. If the controller is stopped and then restarted, IBM Workload Scheduler for z/OS uses the checkpoint data set to return the system to the same state as when it was stopped, ready to continue processing.

IBM Workload Scheduler for z/OS automatically formats the checkpoint data set the first time it is used. In its initial state, the checkpoint data set specifies that a new current plan exists. The new current plan is defined by DD name EQQNCPDS. IBM Workload Scheduler for z/OS attempts to copy the new plan and make it the current plan. If the copy is successful, IBM Workload Scheduler for z/OS is fully operational. If the copy is not successful, IBM Workload Scheduler for z/OS has become active without a current plan.

Note:

1. A strong relationship exists between the IBM Workload Scheduler for z/OS checkpoint data set and the current plan data set. There is also a strong relationship between the event positioning record (EPR) in the checkpoint data set, EQQCKPT, and the tracker event data set, EQQEVDXX, referenced in the controller started task procedure, when a DASD connectivity is used. In fact, the EPR is associated with a specific destination and, therefore, also to a specific event data set. If this relationship is broken, the results of the synchronization processing at controller restart can be unpredictable. This is because events could be lost or reprocessed. Ensure that you do not accidentally delete or overwrite the checkpoint data set
2. To initialize the checkpoint data set, the OPCHOST keyword of the OPCOPTS initialization statement must be set to its default value, that is, OPCHOST(YES), the first time the scheduler is started. With OPCHOST(YES), the NMM initializes the checkpoint data set with FMID and LEVEL corresponding to SSX. The OPCHOST value can then be changed. For example, you can change the value to OPCHOST(PLEX) when the subsystem is used as the controlling system in XCF.

The space allocation for the data set must be at least 15 tracks. This allocation can accommodate 1000 workstation destinations.

Diagnostic data sets (EQQDMSG, EQQDUMP, and SYSMDUMP)

Allocate diagnostic data sets for IBM Workload Scheduler for z/OS address spaces, dialog users, batch jobs, and server.

Diagnostic message and trace data set (EQQDMSG):

You should allocate EQQDMSG for each dialog user. You can allocate EQQDMSG either as a SYSOUT data set or as a DASD data set. Usually only a small volume of diagnostic information exists, so an initial allocation of two tracks of DASD should be enough. If EQQDMSG is not defined, output is written to EQQDUMP.

Diagnostic data set (EQQDUMP):

The tracker, controller, and server write debugging information to diagnostic data sets when validity checking discovers internal error conditions. When diagnostic information is logged, a 3999 user abend normally accompanies it. For service

purposes, always include an EQQDUMP DD statement for every IBM Workload Scheduler for z/OS address space, dialog user, batch job, and server.

Diagnostic data sets are usually allocated as DASD data sets, but they can also be allocated to SYSOUT. Usually only a small volume of diagnostic information exists, so an initial allocation of two tracks on DASD should be enough.

Dump data set (SYSMDUMP):

EQQPCS02 contains two allocations for the SYSMDUMP data set. For an IBM Workload Scheduler for z/OS address space, the data set is allocated with the low-level qualifier SYSDUMP. Allocate a unique SYSMDUMP data set for every IBM Workload Scheduler for z/OS address space. For the scheduler server jobs, SYSMDUMP is allocated with the low-level qualifier SYSDUMPS. EQQPCS01 contains the allocation for the SYSMDUMP data set for IBM Workload Scheduler for z/OS batch jobs; this data set is allocated with the low-level qualifier SYSDUMPB. The IBM Workload Scheduler for z/OS batch jobs can use the same data set. It is allocated with a disposition of MOD in the JCL tailored by EQQJOBS.

Furthermore, SYSMDUMP data sets should be defined with a UACC of UPDATE, that is, WRITE-ENABLED to all user IDs under which a job scheduled by IBM Workload Scheduler for z/OS might possibly be submitted. This is because the SUBMIT SUBTASK of the controller or of the tracker which is submitting a given job might abend while running under the user exit EQQUX001 supplied user ID (RUSER user ID) rather than under the user ID associated with the started task. If this occurs, DUMPTASK fails with an ABEND913 if the user ID in control does not have WRITE access to the SYSMDUMP data set.

The UACC of UPDATE access should be defined to all PIF, dialog, and Dynamic Workload Console servers. If a user is not authorized to update the SYSMDUMP data set, and a server failure occurs while running a request for that user, DUMPTASK fails with an ABEND 912. No diagnostic data will be captured.

Note: If you allocate the SYSMDUMP dataset on your own, consider that the SYSMDUMP allocation can also be changed to use LRECL=4160,RECFM=FBS,BLKSIZE=n*4160 (where "n*4160" is a system-determined multiple block size) according to the new possibilities offered by IPCS and z/OS V1.6 or later. The EQQJOBS allocation has been left unchanged, for compatibility with previous allocated datasets.

Event data sets (EQQEVDS, EQQEVDnn, and EQQHHTP0)

Every IBM Workload Scheduler for z/OS address space requires a unique event data set. The data set is device-dependent and must have only a primary space allocation. Do *NOT* allocate any secondary space. The data set is formatted the first time it is used. Each time you use the data set, IBM Workload Scheduler for z/OS keeps a record of where to start. When the last track of the data set is written, IBM Workload Scheduler for z/OS starts writing on the first track again.

Note: The first time IBM Workload Scheduler for z/OS is started with a newly allocated event data set, an SD37 error occurs when IBM Workload Scheduler for z/OS formats the event data set. Do not treat this as an error.

The data set contains records that describe events created by IBM Workload Scheduler for z/OS job-tracking functions. An event-writer task writes to this data set; an event-reader task reads from it. The job-submit task also uses the event data set to checkpoint its activities, using the first record in the data set (the header record). The submit task in a controller address space takes these checkpoints

when the computer workstation is the same system (the workstation destination is blank), so the address space needs the EQQEVDs event data set allocated even if there is no event writer task. When an event writer task is started in the controller address space, it shares the data set with the submit task.

The header record contains checkpoint information for up to 13 workstations per destination. If you plan to have more than 13 workstations defined to use a single destination, you can allocate the event data set with a large logical record length to accommodate the required number. To calculate the record length required, use this formula:

$$\text{LRECL} = (\text{No-of-WS-with-this-destination} * 6) + 22$$

Because the event data set provides a record of each event, events will not be lost if an event-processing component of IBM Workload Scheduler for z/OS must be restarted. The submit checkpointing process ensures that submit requests are synchronized with the controller, thereby preventing lost requests caused by communication failures.

Define enough space for a single extent data set so that it does not wrap around and write over itself before an event is processed. Two cylinders are enough at most installations. The space allocation must be at least 2 tracks when the record length is 100. There must be sufficient space in the event data set to accommodate 100 records. Consider this requirement if you will define the event data set with a record length greater than 100. For example if you define an LRECL of 15 000, the minimum space allocation is 34 tracks, which equates to 102 records and an event data set that would wrap around very quickly in most installations.

To aid performance, place the event data set on a device that has low activity. If you run programs that use the RESERVE macro, try to allocate the event data set on a device that is not reserved or where only short reserves are taken. The reserve period must be less than 5 minutes.

If you use the job log retrieval function, consider allocating the event data set with a greater LRECL value than that in Table 27 on page 109. This improves performance because input/output (I/O) operations will be reduced because fewer continuation (type NN) events will be created. You can specify 0, or a value from 100 to 32 000 bytes for LRECL. Any other value will cause the event writer to end, and message EQQW053E will be written to the message log. If you do not specify a value for LRECL or specify 0, the data set will be forced to have an LRECL of 100 when it is opened by IBM Workload Scheduler for z/OS. However, the data set must be **unblocked**: the block size must be equal to the logical record length. If you intend to activate job log retrieval function, use one of the these formulas to estimate the LRECL that you should specify:

Calculating the optimum LRECL
 $\text{LRECL} = ((\text{NN}/\text{EV}) * 20) + 100$ OR $\text{LRECL} = (4 * \text{N}) + 100$

In the first formula, NN is the number of continuation events, and EV is the number of all other events. Event types are found in position 21 of the event records. In the second formula, N is the average number of NN events per job. If your calculation yields a value of less than 110, there will be little or no improvement in performance. In this case, you should specify an LRECL value of 100.

You will probably need to test your system first to get an idea of the number and event types that are created. You can then reallocate the event data set when you have gathered information about the events created at your installation. But, before

you reallocate an event data set, ensure that the current plan is completely up-to-date. You must also stop the event writer, and any event reader, that uses the data set.

Note: Do not move IBM Workload Scheduler for z/OS event data sets once they are allocated. They contain device-dependent information and cannot be copied from one device type to another, or moved on the same volume. An event data set that is moved will be reinitialized. This causes all events in the data set to be lost. If you have DFHSM or a similar product installed, you should specify that IBM Workload Scheduler for z/OS event data sets are not migrated or moved.

Event-driven workload automation configuration file data set (EQQEVLIB)

This data set contains the configuration files required by the event-driven workload automation (EDWA) process. The configuration files, which are created by the EQQRXTRG program, are used by the trackers to monitor the event conditions. The event-driven workload automation configuration file data set is accessed by the controller, which, when configuration files are created or modified, deploy them to the trackers by storing the files into the data set identified by the EQQJCLIB DD card. This is the same data set to which the trackers' JCLs refer.

By using the event-driven workload automation configuration file data set, you can automate and centralize the deployment of configuration files to the trackers without having to use the EQQLSENT macro for each tracker.

Job library data set (EQQJBLIB)

The job library data set contains the JCL for the jobs and started tasks that IBM Workload Scheduler for z/OS will submit. It is required by a controller. If you already have a job library that you will use for IBM Workload Scheduler for z/OS purposes, specify this data set on the EQQJBLIB statement. If not, allocate one before you start the controller.

Note: Allocate the job library data set with a only primary space allocation. If a secondary allocation is defined and the library goes into an extent when IBM Workload Scheduler for z/OS is active, you must stop and restart the controller. Also, do not compress members in this PDS. For example, do not use the ISPF PACK ON command, because IBM Workload Scheduler for z/OS does not use ISPF services to read it.

The limitation of allocating the job library data set with only a primary space allocation is not applicable for PDSE data sets.

Note: Each member in the EQQJBLIB must contain one job stream (only one job card), and the job name on the job card must match the job name in the IBM Workload Scheduler for z/OS scheduled operation.

Job-completion-checker data sets

You can optionally use the job completion checker (JCC) to scan SYSOUT for jobs and started tasks. Depending on the JCC functions you want to use, allocate at least one of the three data sets associated with the JCC:

JCC-message-table library (EQQJCLIB):

If the success or failure of a job or started task cannot be determined by system completion codes, the JCC function can be used to scan the SYSOUT created and set an appropriate error code. You determine how the SYSOUT data is scanned by creating JCC message tables. A general message table (EQQGJCCT) must be

defined. Job-specific message tables can be created to search for specific data strings in particular jobs. These tables are stored in the PDS with a member name that matches the job name.

Every IBM Workload Scheduler for z/OS subsystem where you start the JCC task must have access to a message table library. If you want, you can use the same message table library for all IBM Workload Scheduler for z/OS systems.

If you use the data set-triggering function, the data set-selection table (EQQEVLST or EQQDSLST) must be stored in EQQJCLIB.

Note: Allocate the JCC message table data set with only primary space allocation. The limitation is not applicable for PDSE data sets.

JCC-incident-log data set:

You can optionally use the JCC to write records to an incident log data set. This data set is defined by the INCDSN keyword of the JCCOPTS statement.

When scanning SYSOUT data sets, the JCC recognizes events that you define as unusual. If the EQQUX006 exit is loaded by IBM Workload Scheduler for z/OS, the JCC records these events in the incident log data set. The incident log data set can be shared by several JCC tasks running on the same system or on different systems. The data set can also be updated manually or even reallocated while the JCC is active. If the JCC is unable to write to the incident log, the incident work data set is used instead.

JCC-incident work data set (EQQINCWK):

Occasionally, the JCC cannot allocate the incident log data set. This can happen if another subsystem or an IBM Workload Scheduler for z/OS user has already accessed the data set. In this case, the JCC writes to the incident work file, EQQINCWK, instead. If it is not empty, the work file is copied and emptied each time the incident log data set is allocated.

Job-tracking data sets (EQQJTARC, EQQJTnn, EQQDLnn)

Job-tracking data sets are a log of updates to the current plan. They optionally contain audit trail records. Job-tracking data sets comprise:

- Job-tracking logs (EQQJTnn)
- Dual job-tracking logs (EQQDLnn)
- Job-tracking archive (EQQJTARC)

You must allocate EQQJTARC and at least two job-tracking logs (EQQJT01 and EQQJT02) for a controller. The actual number of JT logs that you should allocate is determined by the value that you specify on the JTLOGS keyword of the JTOPTS initialization statement. If you decide to allocate three job-tracking logs, specify the DD names EQQJT01, EQQJT02, and EQQJT03. If you specified EQQJT01, EQQJT02, and EQQJT04, an error occurs and IBM Workload Scheduler for z/OS terminates. IBM Workload Scheduler for z/OS uses the job-tracking logs in turn. When a current plan backup is performed, the active log is appended to EQQJTARC data set.

The size of the CP files, JT and JTARC, can become large, but with appropriate tuning of the size and of the DP frequency, they will not allocate additional extents. If necessary, use the allocation of additional extents (not additional volumes, because only extent allocation is supported in the shipped JT allocation samples). The JTLOG keyword default defines five job-tracking logs. It is recommended that you specify at least three job-tracking logs. Job-tracking logs are

switched at every current plan backup. If the interval between backups is very low and JTLOGS(2) is specified, the previously used job-tracking log might not have been archived before IBM Workload Scheduler for z/OS must switch again. If it cannot switch successfully, the normal-mode-manager (NMM) subtask is automatically shut down, preventing further updates to the current plan.

You can optionally allocate dual JT logs. These logs are identified by the EQQDL nn DD names in the controller started-task JCL. Allocate the same number of dual JT logs as JT logs. The numeric suffixes, nn , must be the same as for the JT logs, because IBM Workload Scheduler for z/OS uses the logs with the same number: EQQJT01 and EQQDL01, EQQJT02 and EQQDL02, and so on. IBM Workload Scheduler for z/OS writes job-tracking information to both logs, so that if the active JT log is lost it can be restored from the dual log, and IBM Workload Scheduler for z/OS can be restarted without losing any events. To achieve the maximum benefit from dual JT logs, you should allocate them:

- With the same attributes as the JT logs
- With at least the same amount of space as the JT logs
- On alternate I/O paths and physical volumes than their corresponding JT logs

IBM Workload Scheduler for z/OS tries to use dual JT logs if you specify DUAL(YES) on the JTOPTS initialization statement of a controller.

The job-tracking-archive data set accumulates all job-tracking data between successive creations of a new current plan (NCP). Therefore, allocate EQQJTARC with enough space for all job-tracking records that are created between daily planning jobs; that is, extend or replan of the current plan. In other words, be sure that you allocate for EQQJTARC an equal or greater amount of space than the total of the space you allocate for the JT files, or you will get a system error. When the daily planning batch job is run, the active job-tracking log is appended to EQQJTARC, and the JT log is switched. The archive log, EQQJTARC, is then copied to the track log data set referenced by the EQQTROUT DD name during the daily planning process. When IBM Workload Scheduler for z/OS takes over the NCP, the archive data set is emptied.

IBM Workload Scheduler for z/OS recovery procedures that use the job-tracking data sets are described in *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Extended-auditing data sets (EQQDBARC, EQQDB nn)

If you have set AUDIT AMOUNT(EXTENDED), extended-auditing data sets are a log of records that show the values that were set before and after the database was changed. They comprise:

- Extended-auditing logs (EQQDB nn)
- Extended-auditing archive (EQQDBARC)

You can control the level of information to be logged by editing the SYSIN card in the EQQAUDIB sample, as follows:

1. Set the input file to use:
 - DBX** For EQQDB nn
 - DBR** For EQQDROUT
2. Specify the level of information to log in the report (this applies only to the delete or add action):
 - K** Only the key is reported.
 - S** Summary information is reported (default).

- F Complete information is reported.
- Set the database to audit. If you do not specify any value, all the databases are audited.
 - AD** Application Description
 - CAL** Calendar
 - JV** Job Variable Table
 - OI** Operator Instructions
 - PER** Period
 - RD** Special Resource
 - RUN** Run cycle group
 - WS** Workstation
 - Specify the database key to audit. If you did not set a database, this value is ignored.
 - ad_name*
For the AD database.
 - calendar_name*
For the CAL database.
 - job_table_name*
For the Job Variable Table database.
 - oi_ad_name oi_op_num*
For the Operator Instructions database.
 - period_name*
For the Period database
 - special_resource_name*
Special Resource
 - run_cycle_group_name*
Run cycle group
 - ws_name*
For the WS database

For example, a SYSIN card to audit the EQQDBnn database by logging the complete information about the workstation named CPU1 in the WS database, looks like the following:

```
//SYSIN DD *
DBXFWS CPU1
/*
```

You must allocate EQQDBARC and at least two extended-auditing logs (EQQDB01 and EQQDB02) for a controller. The actual number of DB logs that you should allocate is determined by the value that you specify on the JTLOGS keyword of the JTOPTS initialization statement. If you decide to allocate three extended-auditing logs, specify the DD names EQQDB01, EQQDB02, and EQQDB03. If you specified EQQDB01, EQQDB02, and EQQDB04, an error occurs and IBM Workload Scheduler for z/OS terminates. IBM Workload Scheduler for z/OS uses the extended-auditing logs in turn. When a current plan backup is performed, the active log is appended to EQQDBARC data set.

The size of the DB and DBARC data sets can become large, but with appropriate tuning of the size and of the DP frequency, they will not allocate additional extents. If necessary, use the allocation of additional extents (not additional volumes, because only extent allocation is supported in the shipped DB allocation samples).

The extended-auditing-archive data set accumulates all extended-auditing data between successive creations of a new current plan (NCP). Therefore, allocate EQQDBARC with enough space for all extended-auditing records that are created

between daily planning jobs; that is, extend or replan of the current plan. In other words, ensure that you allocate for EQQDBARC an equal or greater amount of space than the total of the space you allocate for the DB files, or you will get a system error. When the daily planning batch job is run, the active extended-auditing log is appended to EQQDBARC, and the DB log is switched. The archive log, EQQDBARC, is then copied to the extended-auditing log data set referenced by the EQQDBOUT DD name during the daily planning process. When IBM Workload Scheduler for z/OS takes over the NCP, the EQQDBARC data set is emptied.

IBM Workload Scheduler for z/OS recovery procedures that use the extended-auditing data sets are described in *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Message log data set (EQQMLOG)

The message log data set can be written to SYSOUT or a data set. The data control block (DCB) for this data set is defined by IBM Workload Scheduler for z/OS as follows:

```
EQQMLOG DCB attributes  
DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
```

If the message log data set becomes full during initialization, or when a subtask is restarted, IBM Workload Scheduler for z/OS will abend with error code SD37. In either case, you must stop IBM Workload Scheduler for z/OS and reallocate the message log data set with more space. In all other circumstances, if the data set fills up, IBM Workload Scheduler for z/OS redirects messages to the system log instead.

Note: The scheduler ABENDs with error code sb37 or sd37 if the message log data set becomes full under any of the following circumstances:

- During initialization
- When a subtask is restarted
- While processing any modify command which requires parsing of initialization parameters or specifies the newnoerr, noerrmem(member), or lstnoerr options

In the last case, the ABEND also occurs if the EQQMLOG is already full when any such command is issued. In all these cases you must reallocate more space to the message log data set. In all the other cases, if the data set fills up, the scheduler redirects messages to the system log instead.

EQQPCS02 contains two allocations for the EQQMLOG data set. For an IBM Workload Scheduler for z/OS address space, the data set is allocated with the low-level qualifier MLOG. For the scheduler server jobs, the data set is allocated with the low-level qualifier MLOGS.

Note: If you allocate the message log data set on DASD, define a different data set for IBM Workload Scheduler for z/OS batch program. The data set must also be different from the one used by each IBM Workload Scheduler for z/OS address space (controller, standby controller, tracker, and server). The data set cannot be shared.

See also “Using two message log (MLOG) data sets” on page 42.

Loop analysis log data set (EQQLOOP)

The loop analysis log data set can be written to SYSOUT or a data set. The data control block (DCB) for this data set is defined by IBM Workload Scheduler for z/OS as follows:

EQQLOOP DCB attributes
DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)

This data set is defined the same way as for EQQMLOG, but it is specific for loop analysis and is populated only if a loop condition occurs. It is required by daily planning batch programs (extend, replan, and trial).

Controller and Output collector communication data sets

The EQQOUCV and EQQOUCP data sets are used for the communication that takes place between the controller and Output collector in the retrieval process of the job logs produced in the z-centric environment. The communication process is based upon events. Every time a job in the z-centric environment completes or terminates, the controller queues an event for the output collector with the information necessary to identify the job and the agent that run it.

The communication process is as follows:

1. The Event Manager task of the controller writes an event in EQQOUCV every time a job completes or terminates and updates the next-to-write counter.
2. The Output collector started task reads an event from this data set, checkpoints it in EQQOUCP, dispatches it to the proper thread, and marks the event as processed moving to the next-to-read index in the data set header.

EQQOUCV is a sequential data set organized in records. Includes a header pointing to the next-to-read and next-to-write records.

EQQOUCP is a partitioned data set. It is used to checkpoint the incoming requests to prevent their loss in case of unplanned closures. The EQQOUCV queue manager writes the request in EQQOUCP before placing it in the destination queue in memory while the thread that collects the log from the agent deletes the request after it is satisfied. The member name must be unique but not necessarily meaningful (for example J0000001 or J0000002).

Parameter library (EQQPARM)

Each IBM Workload Scheduler for z/OS subsystem reads members of a parameter library when it is started. Parameter library members (residing in library extent), that have been created, cannot be accessed after they have been opened. To avoid this problem, the data set that defines the EQQPARM library should be allocated without any secondary extents. The limitation is not applicable for PDSE data sets. The library contains initialization statements that define runtime options for the subsystem. Allocate at least one parameter library for your IBM Workload Scheduler for z/OS systems. You can keep the parameters for all your subsystems in one library, as long as it resides on a DASD volume that is accessible by all systems.

PIF parameter data set (EQQYPARM)

Allocate the PIF parameter data set if you intend to use a programming interface to IBM Workload Scheduler for z/OS. The data set can be sequential or partitioned. In the PIF parameter file you specify how requests from the programming interface should be processed by IBM Workload Scheduler for z/OS. By defining an INIT initialization statement in the PIF parameter data set, you override the global settings of the INTFOPTS statement.

The initialization statements are described in *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Automatic-recovery-procedure library (EQQPRLIB)

Allocate a data set for the automatic-recovery-procedure library if you intend to use the IBM Workload Scheduler for z/OS automatic-recovery function. The library is used by the ADDPROC JCL rebuild parameter of the JCL recovery statement. This parameter lets you include JCL procedures in a failing job or started task before it is restarted.

Script library for end-to-end scheduling with fault tolerance capabilities (EQQSCLIB)

This script library data set includes members containing the commands or the job definitions for fault-tolerant workstations. It is required in the controller if you want to use the end-to-end scheduling with fault tolerance capabilities. See *Customization and Tuning* for details on the JOBREC, RECOVERY, and VARSUB statements.

Note: Do not compress members in this PDS. For example, do not use the ISPF PACK ON command, because IBM Workload Scheduler for z/OS does not use ISPF services to read it.

Started-task-submit data set (EQQSTC)

The started-task-submit data set is used by IBM Workload Scheduler for z/OS to temporarily store JCL when a started task is to be started. Use these attributes for this data set:

```
EQQSTC attributes  
SPACE=(TRK,(5,0,1)),  
DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

Include an EQQSTC in the JES PROCLIB concatenation on each system where IBM Workload Scheduler for z/OS schedules started-task operations. The data set is used as a temporary staging area for the started-task JCL procedure. When the start command has been issued for the task and control for the task has passed to JES, IBM Workload Scheduler for z/OS deletes the JCL by resetting the PDS. This means that you never need to compress the data set. For more information, see “Implementing support for started-task operations” on page 129.

Note: IBM Workload Scheduler for z/OS does not support partitioned data set extended (PDSE) libraries for a started-task-submit data set.

Submit/release data set (EQQSUDS)

The submit/release data set is device dependent and must have only a primary space allocation. Do **not** allocate any secondary space. The data set is formatted the first time it is used. Each time you use the data set, IBM Workload Scheduler for z/OS keeps a record of where to start. When the last track of the data set is written, IBM Workload Scheduler for z/OS starts writing on the first track again.

Two cylinders are enough at most installations.

Note:

1. The first time IBM Workload Scheduler for z/OS is started with a newly allocated submit/release data set, an SD37 error occurs when it formats the data set. Expect this, do not treat it as an error.
2. Do not move IBM Workload Scheduler for z/OS submit/release data sets once they are allocated. They contain device-dependent information and cannot be copied from one device type to another, or moved on the same volume. A submit/release data set that is moved will be re-initialized. This causes all

information in the data set to be lost. If you have DFHSM or a similar product installed, define IBM Workload Scheduler for z/OS submit/release data sets so that they are not migrated or moved.

Centralized script data set for end-to-end scheduling with fault tolerance capabilities (EQQTWSCS)

In an end-to-end with fault tolerance capabilities environment, IBM Workload Scheduler for z/OS uses the centralized script data set to temporarily store a script when it is downloaded from the JOBLIB data set to the agent for its submission.

Set the following attributes for EQQTWSCS:

```
DSNTYPE=LIBRARY,  
SPACE=(CYL,(1,1,10)),  
DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

If you want to use centralized script support when scheduling end-to-end with fault tolerance capabilities, you need to use the EQQTWSCS DD statement in the controller and server started tasks. The data set must be a partitioned extended data set.

Input and output events data sets for end-to-end scheduling with fault tolerance capabilities (EQQTWSIN and EQQTWSOU)

These data sets are required by every IBM Workload Scheduler for z/OS address space that uses the end-to-end scheduling with fault tolerance capabilities. They record the descriptions of events related with operations running on fault-tolerant workstations and are used by both the End-to-end enabler task and the translator process in the scheduler's server.

The data sets are device-dependent and can have only primary space allocation. Do not allocate any secondary space. They are automatically formatted by IBM Workload Scheduler for z/OS the first time they are used.

Note: An *SD37* abend code is produced when IBM Workload Scheduler for z/OS formats a newly allocated data set. Ignore this error.

EQQTWSIN and EQQTWSOU are wrap-around data sets. In each data set, the header record is used to track the amount of *read* and *write* records. To avoid the loss of event records, a writer task does not write any new records until more space is available when all the existing records have been read.

The quantity of space that you need to define for each data set requires some attention. Because the two data sets are also used for joblog retrieval, the limit for the joblog length is half the maximum number of records that can be stored in the input events data set. Two cylinders are sufficient for most installations.

The maximum length of the events logged in these two data sets, including the joblogs, is 160 bytes. Anyway, it is possible to allocate the data sets with a longer logical record length. Using record lengths greater than 160 bytes does not produce either advantages or problems. The maximum allowed value is 32000 bytes; greater values will cause the E2E task to terminate. In both data sets there must be enough space for at least 1000 events (the maximum number of joblog events is 500). Use this as a reference, if you plan to define a record length greater than 160 bytes. When the record length of 160 bytes is used, the space allocation must be at least 1 cylinder. The data sets must be unblocked and the block size must be the same as the logical record length. A minimum record length of 160 bytes is necessary for the EQQTWSOU data set in order to be able to decide how to build the job name

in the symphony file (for details about the TWSJOBNAME parameter in the JTOPTS statement, see *Customization and Tuning*).

For good performance, define the data sets on a device with plenty of availability. If you run programs that use the RESERVE macro, try to allocate the data sets on a device that is not, or slightly, reserved.

Initially, you might need to test your system to estimate the number and type of events that are created at your installation. When you have gathered enough information, you can then reallocate the data sets. Before you reallocate a data set, ensure that the current plan is entirely up-to-date. You must also stop the end-to-end sender and receiver task on the controller and the translator thread on the server that use this data set. EQQTWSIN and EQQTWSOU must not be allocated multivolume.

Note: Do not move these data sets once they have been allocated. They contain device-dependent information and cannot be copied from one type of device to another, or moved around on the same volume. An end-to-end event data set that is moved will be re-initialized. This causes all events in the data set to be lost. If you have DFHSM or a similar product installed, you should specify that E2E event data sets are not migrated or moved.

Allocating Data Store data sets

At this stage of your installation, use the EQQPCS04 member generated by the EQQJOBS installation aid. It is contained in the output library specified on the CREATE DATA STORE SAMPLES panel (EQQJOBS5). Submit the EQQPCS04 job to define and initialize the Data Store VSAM files.

Note: You can omit this step if you are migrating from a previous IBM Workload Scheduler for z/OS version.

The Data Store VSAM files can be of three types:

Unstructured

The type associated to EQQUDFxx; used to save joblogs. These files are allocated only when the Joblog Retrieval option in panel EQQJOBS7 is set to Y. The UDF data sets (DDNAME EQQUDFNN) are used only by the data store started task.

Structured

The type associated to EQQSDFxx; used to save structured joblog information. These files are required. The EQQSDFXX data sets allocated for the data store have the same structure as the EQQSDFXX data sets used by the controller but they are a different set.

KSDS The type used for EQQPKIxx and EQQSKIxx. The EQQPKIXX and EQQSKIXX data sets allocated for the data store have the same structure as the structure of the EQQPKIXX and EQQSKIXX data sets used by the controller but they are a different set.

They are listed and described in Table 28 on page 124:

Table 28. Data Store VSAM data sets

Sample	DD Name	Rec. Type	Attributes	Share Option	Keys	Record Size	data set
EQQPCS04	EQQPKIxx	KSDS	UNIQUE INDEXED	1, 3	34 0	77 77	Primary Index
EQQPCS04	EQQSDFxx	LINEAR	N/A	2, 3	N/A	N/A	Data files
EQQPCS04	EQQSKIxx	KSDS	UNIQUE INDEXED	1, 3	38 0	76 32000	Secondary Index
EQQPCS04	EQQUDFxx	LINEAR	N/A	2, 3	N/A	N/A	Data files

For information about how to estimate the size of the Data Store VSAM files, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Allocating data sets for the Dynamic Workload Console reporting feature

Use the EQQPCS09 member generated by the EQQJOBS installation aid and contained in the output library specified on the CREATE SAMPLE JOB JCL panel (EQQJOBS3) to define and allocate:

- The GDG base entry used for old current plan backup which is created during the daily plan batch process, when you specify the BATCHOPTS statement with the JRUNHISTORY parameter set to YES. The GDG data set is identified in the daily planning EXTEND or REPLAN batch job by the EQQOCPBK ddname.
- The VSAM data set where the archiving process copies each generation data set. Allocate the VSAM data set with the same characteristics as the current plan VSAM data set, because it is used to store the old current plan.

For detailed information about the archiving process, see *IBM Workload Scheduler for z/OS: Managing the Workload*.

Allocating the files and directories

The following features use files on UNIX System Services (USS):

- End-to-end scheduling with fault tolerance capabilities.
- End-to-end scheduling with z-centric capabilities, if SSLKEYRINGTYPE is set to USS in the HTTPOPTS statement.
- Features running Java utilities:
 - Historical run data archiving for Dynamic Workload Console reporting
 - Event-driven workload automation for data set triggering

By default, the EQQJOBS installation aid sets the following paths for the following directories:

End-to-end with fault tolerance work directory (EQQJOBS8)

/var/TWS/inst

JAVA utilities enablement work directory (EQQJOBS9)

/var/TWS/inst

SSL for TCP/IP connection work directory (EQQJOBSC)

/var/TWS/inst/ssl

By keeping the default directories, if the end-to-end work directory is deleted, the Java and SSL work directories are also deleted. To avoid this problem, set different paths for the different work directories. For example:

End-to-end with fault tolerance work directory (EQQJOBS8)

/var/TWS/E2E

JAVA utilities enablement work directory (EQQJOBS9)

/var/TWS/JAVAUTL

SSL for TCP/IP connection work directory (EQQJOBSC)

/var/TWS/SSL

To create the correct directories and files, run the following sample jobs for each controller that supports the specific feature:

- The EQQPCS05 sample for the end-to-end scheduling with fault tolerance capabilities
- The EQQPCS08 sample for the historical run data archiving and event-driven workload automation

To run the previous samples, you must have one of the following permissions:

- UNIX System Services (USS) user ID (UID) equal to 0
- BPX.SUPERUSER FACILITY class profile in RACF
- UID specified in the JCL in eqqUID and belonging to the group (GID) specified in the JCL in eqqGID

For the EQQPCS05 sample, if the GID or the UID were not specified in EQQJOBS, you can specify them in the STDENV DD before running the sample. Make sure that you specify a unique UID with a nonzero value; for additional information about this requirement, see INFO APAR II14235.

The user must also have the /bin/sh login shell defined in his OMVS section of the RACF profile. Make sure that the login shell is set as a system default or use the following TSO command to define it:

```
ALTUSER username OMVS(PROGRAM('/bin/sh'))
```

To check the current settings:

1. Run the following TSO command:
LISTUSER *username* OMVS
2. Look in the PROGRAM line of the OMVS section.

After running EQQPCS05, you find the following files in the work directory:

localopts

Defines the attributes of the local workstation (OPCMaster) for batchman, mailman, netman and writer processes and for SSL. The parameters that have no effect in an end-to-end environment are indicated and commented out. For information about customizing this file, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

mozart/globalopts

Defines the attributes of the IBM Workload Scheduler network (OPCMaster ignores them).

Netconf

Netman configuration files

TWSCCLog.properties

Defines attributes for the trace function.

You will also find the following directories in the work directory:

- mozart
- pobox
- stdlist
- stdlist/logs contains the USS processes logs files

After running EQQPCS08, you find the following file in the work directory:

java/env.profile

Defines the environmental variable required by the Java utilities.

It is possible to customize this file and follow the steps listed in section “Setting up to use the DB2 on z/OS for reports on Dynamic Workload Console” on page 128 to enable the use of the DB2 installed on the host for running IBM Workload Scheduler for z/OS reports from the Dynamic Workload Console.

Configuring for end-to-end scheduling with fault tolerance capabilities in a SYSPLEX environment

In a configuration with a controller and no stand-by controllers, define the end-to-end server work directory in a file system mounted under either a system-specific HFS or a system-specificzFS.

Then configure the Byte Range Lock Manager (BRLM) server in a distributed form (see following considerations about BRLM). In this way the server will not be affected by the failure of other systems in the sysplex.

Having a shared HFS or zFS in a sysplex configuration means that all file systems are available to all systems participating in the shared HFS or zFS support. With the shared HFS or zFS support there is no I/O performance reduction for an HFS or zFS read-only (R/O). However, the intersystem communication (XCF) required for shared HFS or zFS might affect the response time on read/write (R/W) file systems being shared in a sysplex. For example, assume that a user on system SYS1 issued a read request to a file system owned R/W on system SYS2. Using shared HFS or zFS support, the read request message is sent via an XCF messaging function. After SYS2 receives the message, it gathers the requested data from the file and returns the data using the same request message.

In many cases, when accessing data on a system which owns a file system, the file I/O time is only the path length to the buffer manager to retrieve the data from the cache. On the contrary, file I/O to a shared HFS or zFS from a client which does not own the mount, requires additional path length to be considered, plus the time involved in the XCF messaging function. Increased XCF message traffic is a factor which can contribute to performance degradation. For this reason, it is recommended for system files to be owned by the system where the end-to end server runs.

In a configuration with an active controller and several stand-by controllers, make sure that all the related end-to-end servers running on the different systems in the Sysplex have access to the same work directory.

On z/OS systems, the shared ZFS capability is available: all file systems that are mounted by a system participating in shared ZFS are available to all participating

systems. When allocating the work directory in a shared ZFS you can decide to define it in a file system mounted under the system-specific ZFS or in a file system mounted under the sysplex root. A system-specific file system becomes unreachable if the system is not active. To make good use of the takeover process, define the work directory in a file system mounted under the sysplex root and defined as automove.

The Byte Range Lock Manager (BRLM) locks some files in the work directory. The BRLM can be implemented:

- With a central BRLM server running on one member of the sysplex and managing locks for all processes running in the sysplex.
- In a distributed form, where each system in the sysplex has its own BRLM server responsible for handling lock requests for all regular files in a file system which is mounted and owned locally (see APARs OW48204 and OW52293).

If the system where the BRLM runs experiences a scheduled or unscheduled outage, all locks held under the old BRLM are lost. To preserve data integrity, further locking and I/O on any opened files is prevented until files are closed and reopened. Moreover, any process locking a file is terminated.

To avoid these kinds of error in the end-to-end server, before starting a scheduled shut down procedure for a system, you must stop the end-to-end server if either or both of the following conditions occurs:

- The work directory is owned by the system to be closed
 - The **df -v** command on OMVS displays the owners of the mounted file systems
- The system hosts the central BRLM server
 - The console command **DISPLAY OMVS,O** can be used to display the name of the system where the BRLM runs. If the BRLM Server becomes unavailable, then the distributed BRLM is implemented. In this case the E2E server needs to be stopped only if the system which owns the work directory is stopped.

The server can be restarted after a new system in the sharing has taken the ownership of the file system and/or a new BRLM is established by one of the surviving systems.

To minimize the risk of filling up the IBM Workload Scheduler internal queues while the server is down, schedule the closure of the system when the workload is low.

A separate file system data set is recommended for each stdlist directory mounted in R/W on `/var/TWS/inst/stdlist`, where *inst* varies depending on your configuration.

When you calculate the size of a file, consider that you need 10 MB for each of the following files: `Intercom.msg`, `Mailbox.msg`, `pobox/tomaster.msg`, and `pobox/CPUDOMAIN.msg`.

You need 512 bytes for each record in the Symphony, Symold, Sinfonia, and Sinfold files. Consider a record for each CPU, schedule, and job/recovery job.

You can specify the number of days that the trace files are kept on the file system using the parameter `TRCDAYS` in the `TOPOLOGY` statement.

Setting up to use the DB2 on z/OS for reports on Dynamic Workload Console

Customize the env.profile file, unloaded in the work directory by EQQPCS08, to use a DB2 on z/OS to run reports from the Dynamic Workload Console without having to install another DB2 on the distributed side.

To use the DB2 on z/OS to run IBM Workload Scheduler for z/OS reports on the Dynamic Workload Console:

1. Open the env.profile file and uncomment and customize the following rows:

```
#export DB2_JDBC_PATH=/usr/lpp/db2910_jdbc/classes
#export CLASSPATH="$DB2_JDBC_PATH"/db2jcc.jar:"$DB2_JDBC_PATH"/db2jcc_license_cisuz.jar:
"$CLASSPATH":
```

where the path specified in the DB2_JDBC_PATH is the path where the db2jcc.jar and db2jcc_license_cisuz.jar files are stored on the z/OS system.

These rows are preceded by the comment:

```
#If DB2 is on z/OS, customize and uncomment the following line
```

2. To establish a connection with the Dynamic Workload Console, send the db2jcc_license_cisuz.jar license jar file from the z/OS system to the Dynamic Workload Console server and store it in the following directories:

```
... \Program Files\IBM\JazzSM\profile\installedApps\JazzSMNode01Cell\isc.ear\TWSWebUI.war\
WEB-INF\lib
```

and

```
... \Program Files\IBM\JazzSM\profile\installedApps\JazzSMNode01Cell\isc.ear\TWSWebUI.war\
WEB-INF\platform\plugins\org.eclipse.birt.report.data.oda.jdbc_4.2.1.v20120820\drivers
```

3. To continue the setup, see the section about how to create the database in the z/OS environment in *Managing the Workload*.

Setting up the Workload Automation Programming Language environment

To create the correct environment to use Workload Automation Programming Language, you must run the EQQWPLCO sample job.

Step 10. Creating JCL procedures for address spaces

Perform this task for a tracker, Data Store, controller or standby controller, output collector.

You must define a JCL procedure or batch job for each IBM Workload Scheduler for z/OS address space.

See “Defining subsystems” on page 83 for details.

The EQQJOBS dialog generates several members in the output library that you specified. The following table lists the members that provide samples for the scheduler’s address spaces:

Table 29. Started task JCL samples for IBM Workload Scheduler for z/OS address spaces

Address Space for:	Member
Controller and tracker	EQQCON (sample started task) EQQCONP (sample started task parameters)
Controller	EQQCONO (sample started task) EQQCONOP (sample started task parameters)

Table 29. Started task JCL samples for IBM Workload Scheduler for z/OS address spaces (continued)

Address Space for:	Member
Tracker	EQQTRA (sample started task) EQQTRAP (sample started task parameters)
Server	EQQSER (sample started task) EQQSERP (sample started task parameters)
Data Store	EQQDST (sample started task) EQQDSTP (sample started task parameters)
Output collector	EQQOUC (sample started task) EQQOUCP (sample started task parameters)

These members contain started task JCL that is tailored with the values you entered in the dialog. Tailor these members further, according to the data sets you require. Alternatively, you can copy a member from the SEQQSAMP library to one of your own libraries, and tailor it manually.

If you create a new library for your IBM Workload Scheduler for z/OS started-task procedures, remember to specify the library in the JES PROCLIB concatenation. Then you must restart JES to include the new library.

If you prefer, you can run IBM Workload Scheduler for z/OS as a batch job rather than as a started task. Here, the JCL can reside in any library and will require a job card, besides the JCL requirements in Table 30 on page 130.

Implementing support for started-task operations

The JCL procedures for started-task operations started by IBM Workload Scheduler for z/OS must be stored in a PDS concatenated on the EQQJBLIB DD name. You can include existing data sets, such as SYS1.PROCLIB, if you prefer. Preparation, tailoring, and variable substitution are handled the same way as for batch job operations. When a started-task operation is started by IBM Workload Scheduler for z/OS, the JCL procedure is written to the started-task-submit data set (EQQSTC) on the system where the operation is to be run. IBM Workload Scheduler for z/OS issues a START command for this procedure and then removes the JCL procedure from the EQQSTC data set.

JES2 users should specify the started-task-submit data set on the PROC nn DD statement of the JES2 JCL procedure on each z/OS system. The suffix nn is the value specified for the PROCLIB parameter of the STCCCLASS statement in JES2PARM. To ensure that the correct version of the JCL procedure is started, place the EQQSTC data set first in the concatenation.

JES3 users should specify the started-task-submit data set on the IATPLB nn DD statement of the JES3 global system. The suffix nn is the value specified in the JES3 standards parameter STCPROC. To ensure that the correct JCL procedure will be started, place the EQQSTC data set first in the concatenation. For each submit task that is running on a JES3 local system in the JES3 complex, also include that data set in the JES3 global concatenation.

If you do not use the Restart and Cleanup function, you must follow the previous instructions to work with started-task operations. Otherwise, because the Restart and Cleanup function adds a job card to the procedures for scheduled STC workstation operations at the same time that it adds the //TIVDSTxx output JCL

statements, there are some exceptions to the previous instructions if you want to use the Restart and Cleanup function. The JCL for a started task can contain a job card *only* if the JCL is in a data set in the IEFPDSI or IEFJOBS concatenations of MSTJCLxx when the start command is issued.

You must add the EQQSTC data set to the IEFPDSI DD statement in MSTJCLxx instead of to the JES2 PROCnn or the JES3 global IATPLBnn DD statement as mentioned above.

In addition, all data sets listed in IEFPDSI must be included in the system master catalog.

Note:

1. To include EQQSTC, you must restart JES.
2. Do not use the BLDL parameter of the JES3 PROC statement to specify the procedure name of a started task that is to be scheduled by IBM Workload Scheduler for z/OS.

The EQQSTC data set can be shared by IBM Workload Scheduler for z/OS subsystems that run on the same z/OS image. If you use *global resource serialization* (GRS), the EQQSTC data set can be shared by all z/OS systems defined in the GRS ring if you propagate requests for the resource. To propagate the resource requests to all systems in the ring, define the resource SYSZDRK.data data set name in the SYSTEM inclusion RNL of the GRSRNLnn member of SYS1.PARMLIB. For more information about defining the GRS resource name list, see *z/OS Initialization and Tuning Reference*.

Required data sets

Table 30 shows the data sets required by an IBM Workload Scheduler for z/OS started task. Include the data sets in your JCL procedures as indicated in this table.

Table 30. IBM Workload Scheduler for z/OS required data sets

DD Name	Required by				Defines
	Controller	Tracker	Server	Data Store	
EQQADDS	✓				Application descriptions and JCL variable tables
EQQBRDS	✓	✓			A JES internal-reader
EQQCKPT	✓				Checkpoint data set
EQQCP1DS	✓				Primary current plan
EQQCP2DS	✓				Alternate current plan
EQQCXDS	✓				Current plan extension
EQQEVDS	✓	✓			Event data set for the submit checkpointing function and for the event-writer task
EQQEVLIB	✓				Configuration file repository for event-triggered resource handling
EQQJBLIB	✓				JCL PDS libraries
EQQLOGRC	✓				Joblog and Restart Information pending requests log data set
EQQJS1DS	✓				Primary JCL repository
EQQJS2DS	✓				Alternate JCL repository

Table 30. IBM Workload Scheduler for z/OS required data sets (continued)

DD Name	Required by				Defines
	Controller	Tracker	Server	Data Store	
EQQITABL	✓				Job table log file. The scheduler considers this data set as required only if you defined at least one critical job. Allocate it with the same size as EQQJTARC.
EQQITARC	✓				Job-tracking archive
EQQJTnn	✓				Job-tracking logs
EQQLTDS	✓				Long-term plan
EQQMLIB	✓	✓	✓	✓	Message library
EQQMLOG	✓	✓	✓	✓	Output message log
EQQNCPDS	✓				New current plan
EQQNCXDS	✓				New current plan extension
EQQNXDDS	✓				NCP extension
EQQOIDS	✓				Operator instructions
EQQPARM	✓	✓	✓	✓	Parameter library
EQQRDDS	✓				Special resource descriptions
EQQSCPDS	✓				Current plan backup copy data set for the creation of Symphony. Needed for integration with IBM Tivoli Monitoring.
EQQSIDS	✓				Side information; ETT criteria and configuration data
EQQWSDS	✓				Workstation, calendar and period descriptions
EQQXD1DS	✓				CP1 extension
EQQXD2DS	✓				CP2 extension

Note:

1. The data sets that are required for a controller are also required for a standby controller.
2. The number of job-tracking-log data sets to include depends on the value that you specify in the JTLOGS keyword of the JTOPTS initialization statement. Specify at least 3 job-tracking logs. The default value is 5.
3. You must specify EQQEVDs for a controller even if an event writer is not started in the controller address space. The EQQEVDs data set is used for submit checkpointing. It can be the same data set that is used by an event-writer function. Use a unique EQQEVDs for each address space.
4. In order to set the TCP/IP task up correctly, you need to change the scheduler start procedure to include the C runtime libraries (CEE.SCEERUN in the STEPLIB DD statement).

If you have multiple TCP/IP stacks, or if the name you used for the procedure that started the TCPIP address space was not the default (TCPIP), then you must change the start procedure to include the SYSTCPD DD card to point to a data set containing the TCPIPJOBNAME parameter.

The standard method to determine the connecting TCP/IP image is:

- Connect the TCP/IP specified by TCPIPJOBNAME in the active TCPIP.DATA

- Locate TCPIP.DATA using the SYSTCPD DD card.

Optional data sets

Table 31 shows the data sets that you can optionally include in your JCL procedures. Specify these data sets only if you want to use the function with which they are associated.

Table 31. IBM Workload Scheduler for z/OS optional data sets

DD Name	Can be used by					Defines
	Controller	Tracker	Server	Data Store	Output collector	
AUDITPRT	✓					Input to EQQAUDIT
EQQDLnn	✓					Dual job-tracking logs
EQQDUMP	✓	✓	✓			Diagnostic dump output
EQQEVDnn	✓					Event data set for an event-reader task
EQQINCWK		✓				JCC incident work file
EQQJCLIB		✓				JCC library for message tables and for data set triggering selection table
EQQMONDS	✓					Data set used by monitoring task to store events for IBM Tivoli Monitoring.
EQQOUCFV	✓				✓	Data set used for the communication that takes place between the controller and Output collector in the retrieval process of the job logs produced in the z-centric environment.
EQQOUCKP	✓				✓	Data set used for the communication that takes place between the controller and Output collector in the retrieval process of the job logs produced in the z-centric environment.
EQQPKlxx	✓					Primary index
EQQPRLIB	✓	✓				Automatic-recovery procedures
EQQSCLIB	✓					Scriptlib
EQQSDFnn	✓					Structured data files
EQQSKlxx	✓					Secondary index
EQQSTC	✓	✓				Started-task-submit data set
EQQSUDS		✓				Submit/release data set for an event-writer task
EQQTROUT	✓					Input to EQQAUDIT
EQQTWSCS	✓		✓			Data set for centralized script support in end-to-end scheduling with fault tolerance capabilities
EQQTWSIN	✓		✓			Input event data set in end-to-end scheduling with fault tolerance capabilities
EQQTWSOU	✓		✓			Output event data set in end-to-end scheduling with fault tolerance capabilities

Table 31. IBM Workload Scheduler for z/OS optional data sets (continued)

DD Name	Can be used by					Defines
	Controller	Tracker	Server	Data Store	Output collector	
EQQUDFnn				✓		Unstructured data files
STDENV			✓			This data set/member contains the environment variables of the end-to-end with fault tolerance capabilities processes
STEPLIB	✓	✓	✓			Load-module library
SYSMDUMP	✓	✓	✓			Dump data set
<i>user-defined</i>	✓					Submit/release data set for the controller submit task

Note:

1. The optional data sets that you specify for a controller must also be specified for a standby controller.
2. If you use dual job-tracking, the number of dual job-tracking logs (EQQDLnn) must be the same as the number of job-tracking logs (EQQTmm).
3. Include EQQDUMP and SYSMDUMP for diagnostic purposes.
4. The EQQEVDnn DD name identifies the event data set for an event-reader task. The nn value is the sequence number specified in the ERSEQNO keyword of the event reader (controller only) that will process this data set. It is always a 2-digit number. That is, if the sequence number is less than 10, a leading 0 must be added.
5. Specify the EQQSTC data set if you use IBM Workload Scheduler for z/OS to schedule started-task operations.
6. Use the standard JCL naming conventions for each user-defined DD name; that is, 1-8 alphanumeric or national characters, of which the first character must be alphabetic or national.
7. The submit/release data set is identified by a controller, with a user-defined DD name. The same name must appear in the procedure JCL, the DASD keyword of the ROUTOPTS statement, and the destination field of the workstation representing the system that work is to be sent to. The same data set is identified in a tracker, by the EQQSUDS DD name.
8. When using end-to-end functions, the same EQQTWSIN, EQQTWSOU, and EQQTWSCS data sets must be allocated to the controller and the end-to-end server.
9. The STDENV DD name can point to a sequential DS or a PDS member (for example, a member of the PARMLIB) in which the user can define environment variables to initialize Language Environment®. STDENV must have a F or FB format with a record length equal or greater than 80. In this data set/member you can put your environment variables specifying VARNAME=value. On each row you can specify only 1 variable, characters after column 71 are ignored. If you need more than 71 characters, you can add any character in column 72 and continue on the next row (the character in column 72 is ignored).
10. THE EQQTROUT DD card must point to a data set or be dummy, but it cannot be removed from the daily plan jcl. In particular if a cp extend or replan job is submitted with the EQQTROUT DD deleted or commented out, the DRTOP/DNTOPT JOBSTEP can end with RC08, even if a new plan is

created and taken over, because EQQTROUT could not be opened. Also the data set pointed by the EQQTROUT DD must be allocated by using the DCB values provided in the allocation JCL sample:
RECFM=VB,LRECL=32756,BLKSIZE=32760 otherwise the contents of the data set will be unreadable.

For detailed information about configuring the environment variables for SSL protocol, see how to enable the FIPS compliance over the SSL secured connection in *Scheduling End-to-end with Fault Tolerance Capabilities*.

11. Data sets EQQOUCV and EQQOUCKP are required if you activate the Output collector started task.

Step 11. Defining the initialization statements

In this step of your installation, you define the initialization statements.

When IBM Workload Scheduler for z/OS starts running, it reads the parameter library to determine initialization options and parameters. The parameter library is specified by the PARM parameter of the EXEC statement for the tracker and controller started-task procedures, and by the EQQPARM DD statement for all the other started tasks and EQQBATCH jobs. If the PARM parameter is not specified, the default member name STDPARMS is used; if the name does not exist, message EQQZ010E is issued.

The initialization statements that you should define depend on the functions of IBM Workload Scheduler for z/OS that you want to use. For details about how to define initialization statements, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Step 12. Creating the DB2 database

In this optional step, required only if you need the history function, you create the DB2 database, tables, and indexes. If you need to migrate the history data that you have already defined, then run only the EQQICNVH sample. You can use the history function to rerun operations after they have completed and are no longer in the current plan. When the history function is active, details about completed operations are copied to the DB2 database when you extend the current plan, and remain in the database for a period that you specify. See *IBM Workload Scheduler for z/OS: Managing the Workload* for a description of how to use the history function, and *IBM Workload Scheduler for z/OS: Customization and Tuning* for a description of the necessary initialization parameters.

Note: Do not stop DB2 with the quiesce option if the scheduler history function is implemented. A quiesce of DB2 could cause any dialog user to hang until DB2 termination is complete. Use the +stop db2, force command, instead.

Edit and run the supplied sample job EQQINIDB. This job:

1. Binds the DB2 plan.
2. Grants authorities.
3. Creates the database.
4. Creates the tablespace.
5. Creates the tables and indexes.

Save the output of this job, because this lists the objects that were created, with their parameters. Ensure that you bind the plan and grant the necessary authorities after applying service to IBM Workload Scheduler for z/OS.

Sample to migrate the history database

Use the EQQICNVH sample to migrate the IBM Workload Scheduler for z/OS Operation History data from one release to another.

EQQICNVH is a job with the following steps:

1. **IDCAMS**

Makes it possible to rerun the job by deleting the data sets created in the previous run.

2. **UNLOAD**

Uses the DB2 utility named DSNTIAUL to unload the four tables with the Operation History data, the main table OPCMAIN, the occurrence table OPCOCC, the operation table OPCOPR and the joblog table OPCJL into the UNLOAD data sets. DSNTIAUL also creates the control statements required to later load the unloaded tables in the x.HISTMIG.CNTL data set.

3. **EDIT**

Modifies the LOAD control statements created in the preceding step to make the LOAD step add table entries to the existing tables instead of replacing the tables. The RESUME YES parameter is added.

4. **CONVERT**

Reads the entries of the unloaded tables. You specify the input release and the input subsystem name, and the output release and the output subsystem name. The entries in the UNLOAD data sets for the input release and input subsystem are converted to the record layouts of the output release and given the output subsystem name. The converted table entries are written to the LOAD data sets. The input and output releases can be the same.

5. **LOAD**

Uses the DB2 LOAD utility to add the new table entries to the Operation History tables; that is, the entries created by the CONVERT step.

6. **RESET**

Uses the DB2 REPAIR utility to clean the Operation History tables after the processing of the LOAD step. This step is necessary because LOG NO is specified in the DB2 LOAD utility control statements.

After the job successfully run, the following data sets are created:

- x.UNLOAD.EQQHIMN
- x.UNLOAD.EQQHI3P
- x.UNLOAD.EQQHI3C
- x.UNLOAD.EQQHI14
- x.LOAD.EQQHIMN
- x.LOAD.EQQHI3P
- x.LOAD.EQQHI3C
- x.LOAD.EQQHI14
- x.HISTMIG.CNTL

After a successful run, the UNLOAD data sets contain a backup of the Operation History tables.

The statements in data set x.HISTMIG.CNTL can be used to re-create the original contents of the Operation History tables. Remove the RESUME YES parameters.

At the end of the process, you must make a BIND of the new version. You can use the BIND step provided with the EQQINIDB sample.

Step 13. Setting up the ISPF environment

Perform this task if you are installing the scheduler dialogs.

Because IBM Workload Scheduler for z/OS dialogs run under ISPF, you must set up an ISPF environment. If you are not familiar with ISPF dialogs, see *ISPF Guide and Reference* and *ISPF Examples*.

To set up your ISPF environment, perform these steps:

1. Set up the IBM Workload Scheduler for z/OS CLIST library.
2. Set up the ISPF tables.
3. Allocate ISPF and IBM Workload Scheduler for z/OS data sets to the TSO session.
4. Invoke the IBM Workload Scheduler for z/OS dialog.

These steps are described in the following sections.

Setting up the CLIST library

When you ran the SMP/E apply job, the scheduler CLIST library was copied to a data set allocated to DD name SEQQCLIB. Allocate this data set to the SYSPROC DD name of the TSO logon procedure JCL. This library includes the EQQXSUBC CLIST, which is used by the IBM Workload Scheduler for z/OS dialog when a user requests an IBM Workload Scheduler for z/OS background batch job to be submitted.

For the online EQQAUDIT to work, either copy EQQAUDNS into a library that is part of the TSO SYSPROC concatenation or add the batch-job skeleton library, which is created by EQQJOBS, into the SYSPROC concatenation.

Setting up the ISPF tables

These are the tables in the SEQQTBL0 library that you must allocate to the ISPF table library (ISPTLIB):

EQQACMDS

ISPF command table

EQQAEDIT

Default ISPF edit profile

EQQELDEF

Default ended-in-error-list layouts

EQQEVERT

Ended-in-error-list variable-entity read table

EQQLUDEF

Default dialog connect table

EQQRLDEF

Default ready-list layouts

EQQXVART

Dialog field definitions

If you use the ISPF command table EQQACMDS, invoke IBM Workload Scheduler for z/OS as a separate ISPF application with the name EQQA. “Invoking the IBM Workload Scheduler for z/OS dialog” on page 140 describes this in more detail. If you want to use a different ISPF application name, for example EQQB, create a command table with the name EQQBCMDS.

The customization of the ISPF Dialog is affected and depends on the ISPF application names. This makes necessary that you create copies of the EQQACMDS and EQQAEDIT members of SEQQBTL0 for each ISPF application and locate these copies in ISPTLIB. For example, for the ISPF application names EQQX and EQQY you need to create the ISPTLIB members EQQXCMDs, EQQYCMDs, EQQXEDIT, and EQQYEDIT.

If necessary, you can modify or create an ISPF command table, using ISPF/PDF option 3.9. Note that ISPF/PDF option 3.9 writes the created or modified table to the data set allocated to the ISPTABL.

Setting up the default dialog-controller connection table

Table EQQLUDEF contains values used when establishing the connection between the scheduler dialog user and the controller. These are default values set initially for your installation by the system programmer. Individual users can then modify the values to suit their requirements. Modify the table, adding the following information:

- The names of the controllers in your installation
- When a controller is accessed remotely, the combination of the controller name and the LU name of a server set up to communicate with it
- The set of dialog-controller connections that are to be available to all dialog users

When a user opens the scheduler dialog 0.1, the scheduler first tries to read the connection table EQQALTCP in the ISPF profile library ISPPROF. The connection table name begins with the NEWAPPL ID specified when invoking the scheduler dialog. For example, if the ISPF application name is EQQB, the connection table name is EQQBLTCP. If you used a different ISPF application name *xxxx*, the connection table name is *xxxxLTCP* (if the application name is shorter than four characters, it is filled with *x* up to length 4). If it cannot find the table, it reads the default connection table EQQLUDEF from the ISPTLIB allocation.

When a user modifies the connection table (through the scheduler dialog option 0.1), the changes are written to the EQQALTCP (or *xxxxLTCP*) table of ISPPROF.

To change the distributed EQQLUDEF table:

1. Choose the scheduler dialog option 0.1.
2. Set up the dialog-controller connections for the installation.
3. Copy the connection table EQQALTCP (or *xxxxLTCP*) from your ISPF profile library to the scheduler table library allocated to ISPTLIB, renaming the copy to the default connection table name EQQLUDEF.
4. Optionally, to change the default subsystem name value shown on the EQQOPCAP panel, you need to change the default setting of the `&xopcnm` variable (currently set to OPCC) in the EQQXINIP dummy panel. You must do this before the 0.1 dialog option is selected for the first time, so that your customized value is used when the EQQLUDEF profile is read and the EQQxLTCP profile is created upon the first access to dialog option 0.1.

You can access and work with different controllers from the same TSO session, using ISPF SPLIT to start different IBM Workload Scheduler for z/OS instances with different ISPF application names. In this case you might want to add more than one option to invoke IBM Workload Scheduler for z/OS from the ISPF master application menu, as in the following example:

```

BODY
.
.
.
  1 ..... - .....
  2 ..... - .....
  ..... - .....
  OA OPC - Operations Planning and Control A <===
  OB OPC - Operations Planning and Control B <===
  ..... - .....
PROC
.
.
.
  1, ....
  2, ....
  .., ....
  OA, 'PANEL(EQQOPCAP) NEWAPPL(EQQA)
  OB, 'PANEL(EQQOPCAP) NEWAPPL(EQQB)
.
.
.
END

```

Note: Because the value of the ISPF variable &XOPCNM. (displayed in the EQQOPCAP dialog as "You are communicating with xxxx") and the default controller selected in the 0.1 dialog (EQQXLUSL) are stored, respectively, in members xxxxPROF and xxxxLOUT, make sure that any changes you make to these ISPF profile members are made consistently. For example, if you modify or delete xxxxPROF, you must also modify or delete xxxLOUT.

Setting up list tables and graphical attribute tables

The ISPF tables for list layouts, EQQRLDEF and EQQELDEF, are the default tables displayed for all IBM Workload Scheduler for z/OS dialog users in your installation. They can be modified to suit an individual user's requirements or you can create new defaults for all users in your installation. Modified tables are stored in the user's ISPF profile library under another member name. *IBM Workload Scheduler for z/OS: Customization and Tuning* describes how to modify the default tables for your installation.

GDDM default values are used for graphical attributes. The defaults can be modified to suit the requirements of an individual user or you can create default values for all users. Modified defaults are stored in the EQQAXGRC member of the ISPF profile data set.

When setting up these tables for dialog users, keep the following points in mind:

- When a user requests a graphical display using the GRAPH command, IBM Workload Scheduler for z/OS first searches through the ISPPROF library for the EQQAXGRC ISPF table. If it cannot find the table there, the product searches the ISPTLIB library for the table.
- When a user modifies the graphical display attributes (using the ATTR command from within an IBM Workload Scheduler for z/OS dialog), the EQQAXGRC ISPF table is written to the ISPPROF library.
- When a user displays an ended-in-error list, IBM Workload Scheduler for z/OS first searches for the layout in the EQQELOUT table on ISPPROF. If it cannot find the layout there, the product uses the layout from the EQQELDEF table on ISPTLIB.
- When a user modifies an ended-in-error list layout, the changes are written to the EQQELOUT table.

- When a user displays a ready list, IBM Workload Scheduler for z/OS first searches for the layout in the EQQRLOUT table of ISPPROF. If it cannot find the layout there, the product uses the layout from the EQQRLDEF table on ISPTLIB.
- When a user modifies a ready list layout, the changes are written to the EQQRLOUT table.

Allocating dialog data sets to your TSO session

Table 32 describes the ISPF and IBM Workload Scheduler for z/OS data sets that you must allocate to the TSO session to run the IBM Workload Scheduler for z/OS dialog.

Table 32. ISPF and IBM Workload Scheduler for z/OS dialog data sets

DD Name	IBM Workload Scheduler for z/OS use	Created by
SYSPROC	CLIST library	SMP/E run (SEQQCLIB)
ISPPROF	User-session defaults, read/write tables	Your existing ISPPROF data set
ISPLLIB	Panel library	SMP/E run (SEQQPxxx, SEQQGxxx)
ISPLMLIB	Message library	SMP/E run (SEQQMxxx)
ISPSLIB	Skeleton JCL library	EQQJOBS option 2
ISPTLIB	Read tables (default)	SMP/E run (SEQQTBL0)
EQQMLIB	Message library	SMP/E run (SEQQMxxx)
EQQMLOG	Message log	TSO logon procedure
EQQTMP	Advanced ISPF panel templates	SMP/E run (SEQQLxxx)

Note:

1. The *xxx* suffix represents the national language version supplied with your distribution tape.
2. If you did not install the IBM Workload Scheduler for z/OS load modules in a library defined in the LNKLST*nn* member of SYS1.PARMLIB, also allocate the load-module library to either the STEPLIB or ISPLLIB DD statements. Except for the EQQMINOM module, the IBM Workload Scheduler for z/OS dialog modules need not run APF-authorized. So if EQQMINOM is not in the LNKLST*nn* concatenation, you must copy it to another library so that it can be loaded APF-authorized. The product dialog loads EQQMINOM through IKJEFTSR, therefore you cannot use LIBDEF to add the library containing EQQMINOM to your STEPLIB or ISPLLIB concatenations.
3. Consider allocating EQQDMSG and EQQDUMP to the TSO session for diagnostic purposes.
4. Ensure that the library containing IBM Workload Scheduler for z/OS batch job skeletons, generated by EQQJOBS, is allocated to the ISPSLIB DD statement.
5. You need the EQQMLIB library to run the IBM Workload Scheduler for z/OS TSO commands or to use a TCP/IP connected dialog server.
6. The EQQMLOG data set must be allocated to the TSO session of ISPF dialog users to ensure catching all the AUDIT related error messages when the interactive invocation of the audit is used.

7. For the online EQQAUDIT to work, either copy EQQAUDNS into a library that is part of the TSO SYSPROC concatenation or add the batch-job skeleton library, which is created by EQQJOBS, into the SYSPROC concatenation.
8. EQQTmpl identifies the libraries where the Advanced ISPF panel templates are loaded. The templates are the predefined layouts available for the advanced ISPF panels.

More views are provided for the same panel, for example, for the EQQMOPRV panel (list of operations in the plan), the templates provided are:

EQQMOPRT	Compact view
EQQMOPLT	Full view
EQQMOPJT	Job Detail view

Invoking the IBM Workload Scheduler for z/OS dialog

The following section outlines ways of invoking the IBM Workload Scheduler for z/OS dialog.

Using the EQQOPCAC sample CLIST

You can invoke the IBM Workload Scheduler for z/OS dialog by using the sample CLIST EQQOPCAC. When you run the sample CLIST in TSO READY mode, EQQOPCAC allocates the dialog data sets and invokes ISPF with the initial master panel EQQ@MSTR. The EQQ@MSTR panel, which is in the IBM Workload Scheduler for z/OS panel library, lets you select the applications ISPF/PDF or IBM Workload Scheduler for z/OS.

Modifying an existing ISPF selection menu

You can invoke the IBM Workload Scheduler for z/OS dialog by including IBM Workload Scheduler for z/OS as an option on your existing ISPF master application menu, or on any other selection menu. The following example shows how to do this. The statements that you insert are marked on the right with an arrow (<====).

```
ISPF-selection-menu modification for IBM Workload Scheduler for z/OS
)BODY
:
  1 ..... - .....
  2 ..... - .....
  . ..... - .....
  0 OPC - Operations Planning and Control <====
  . ..... - .....
)PROC
:
REQCLEANUP - Created by ActiveSystems 12/14/99
Entity not defined.
= TRANS(TRUNC(REQCLEANUP - Created by ActiveSystems 12/14/99
Entity not defined.,'.')
  1 , ....
  2 , ....
  . , ....
  0 , 'PANEL(EQQOPCAP) NEWAPPL(EQQA)' <====
  . , ....
:
)END
```

Before you can invoke the IBM Workload Scheduler for z/OS dialog, allocate the data sets. You can allocate these data sets through the TSO logon procedure, or by running a CLIST after TSO logon.

Although you can use any name that follows the guidelines already established at your installation, the sample ISPF command table, `EQQACMDS`, is valid only if you use the ISPF application name `EQQA`. If you change the application name on the `ISPSTART` command, remember to create the corresponding ISPF command table in the table library.

Selecting the main menu directly from TSO

You can invoke the IBM Workload Scheduler for z/OS dialog by selecting the main menu directly from TSO. You do this from TSO by entering this TSO command:

```
/*Invoking the IBM Workload Scheduler for z/OS dialog directly from TSO*/  
ISPSTART PANEL(EQQOPCAP) NEWAPPL(EQQA)
```

Using this method to invoke the dialog means that the main menu, panel `EQQOPCAP`, is the first ISPF panel displayed. If you enter the ISPF command `SPLIT`, `EQQOPCAP` is displayed on the alternate screen. With this method, you cannot use ISPF/PDF and IBM Workload Scheduler for z/OS dialogs at the same time. This method is therefore suitable for users who require *only* IBM Workload Scheduler for z/OS.

Using the ISPF select service

You can invoke the IBM Workload Scheduler for z/OS dialog by using the `SELECT` command from a CLIST or from a program. See your ISPF publications to review these procedures.

Switching to the advanced style for ISPF panels

To use the advanced style for ISPF panels, you need to specify `Y` in the `0.8` option, `SETTING PANEL STYLE`. The advanced ISPF panels enable you to get a quick, at-a-glance scrollable view of the AD and CP operations, with color-coded fields that represent application and operation status, as well as the addition of an Action menu from where you can select administrative tasks to perform. They are provided for the AD application to enable you to list and browse a single AD and also for the CP operation to list and browse a single operation in the plan. All of the commands available for an operation in the current plan are concentrated in the new operation list panel (`EQQSOPRV`, `EQQMOPRV`).

Step 14. Using XCF for communication

Include this task when installing a tracker, controller, standby controller, or Data Store that will use XCF for communication.

If you want to use the cross-system coupling facility (XCF) for communication between IBM Workload Scheduler for z/OS systems, you must:

- Ensure that XCF startup options are suitable for your IBM Workload Scheduler for z/OS configuration
- Include the necessary initialization-statement options for each IBM Workload Scheduler for z/OS started task.

XCF groups

An IBM Workload Scheduler for z/OS XCF system consists of one controller and one or more trackers defined as members in the XCF group. You can include one or more standby controllers in the group. If you want to connect the Data Store to the controller via XCF, you need to define a specific XCF group for them, different to the one defined to connect the controller to the z/OS tracker. You can also

specify more than one IBM Workload Scheduler for z/OS group in a sysplex. For example, you might want to have a test and production IBM Workload Scheduler for z/OS group in your sysplex.

IBM Workload Scheduler for z/OS supports these sysplex configurations:

MULTISYSTEM

XCF services are available to IBM Workload Scheduler for z/OS started tasks residing on different z/OS systems.

MONOPLEX

XCF services are available only to IBM Workload Scheduler for z/OS started tasks residing on a single z/OS system.

Note: Because IBM Workload Scheduler for z/OS uses XCF signaling services, group services, and status monitoring services with permanent status recording, a *couple* data set is required. IBM Workload Scheduler for z/OS does not support a *local sysplex*.

With XCF communication links, the controller can submit workload and control information to trackers that use XCF signaling services. The trackers use XCF services to transmit events to the controller. IBM Workload Scheduler for z/OS systems are either ACTIVE, FAILED, or NOT-DEFINED for the IBM Workload Scheduler for z/OS XCF complex.

Each active member tracks the state of all other members in the group. If an IBM Workload Scheduler for z/OS group member becomes active, stops, or terminates abnormally, the other active members are notified. This list describes the actions taken by each started task in the group:

controller

When the controller detects that a tracker member changes to failed state, it stops sending work to the tracker. When it detects that a tracker has become active, it sends work to the tracker system and instructs the tracker to start transmitting event information.

Standby

When a standby controller that is enabled for takeover detects that the controller has changed to failed state, it attempts to become the new controller. If there is more than one standby controller in the group, the first one to detect failure of the controller attempts to take over the controller functions.

tracker

When a tracker member detects that the controller or standby controller has failed, it stops sending event information. The tracker member continues to collect events and writes them to the event data set. When the controller or standby controller becomes active again it informs the tracker that it is ready to receive events.

XCF runtime options

You specify XCF runtime options in the COUPLE nn member of SYS1.PARMLIB and change them using SETXCF operator commands. “Updating XCF initialization options” on page 89 describes how to change the options in the COUPLE nn member.

Initialization statements used for XCF

IBM Workload Scheduler for z/OS started tasks use these initialization statements for XCF for controller/tracker connections:

XCFOPTS

Identifies the XCF group and member name for the started task. Include XCFOPTS for each started task that should join an XCF group.

ROUTOPTS

Identifies all XCF destinations to the controller or standby controller. Specify ROUTOPTS for each controller and standby controller.

TRROPTS

Identifies the controller for a tracker. TRROPTS is required for each tracker on a controlled system. On a controlling system, TRROPTS is not required if the tracker and the controller are started in the same address space, or if they use shared DASD for event communication. Otherwise, specify TRROPTS.

IBM Workload Scheduler for z/OS started tasks use these initialization statements for XCF for controller/Data Store connections:

CTLMEM

Defines the XCF member name identifying the controller in the XCF connection between controller and Data Store.

DSTGROUP

It defines the XCF group name identifying the Data Store in the XCF connection with the controller.

DSTMEM

XCF member name, identifying the Data Store in the XCF connection between controller and Data Store.

DSTOPTS

Defines the runtime options for the Data Store.

FLOPTS

Defines the options for Fetch Job Log (FL) task.

XCFDEST

It is used by the FL (Fetch Job Log) task to decide from which Data Store the Job Log will be retrieved.

If you did not include these runtime options when you defined the initialization statements, do this now. "Step 11. Defining the initialization statements" on page 134 and *IBM Workload Scheduler for z/OS: Customization and Tuning* describe the initialization statements.

Step 15. Activating the network communication function

Include this task when installing a tracker, controller, or standby controller that will use NCF for communication.

If you want to use a VTAM link to connect a tracker to the controller, activate NCF. The controller can then send work to the tracker and receive event information back, using the VTAM link. To achieve this connection, activate NCF in both the controller and the tracker. To do this:

- Add NCF to the VTAM network definitions.
- Add NCF session parameters.

- Activate network resources.

If you want to connect a controller and Data Store using SNA you need different VTAM definitions. NCF is involved only in the tracker connection; the equivalent task in the Data Store connection is the FN task.

Adding NCF to the VTAM network definitions

You must define NCF as a VTAM application on both the controlling system and each controlled system. Before defining NCF, select names for the NCF applications that are unique within the VTAM network.

To define NCF as an application to VTAM:

1. Add the NCF applications to the application node definitions, using APPL statements.
2. Add the application names that NCF is known by, in any partner systems, to the cross-domain resource definitions. Use cross-domain resource (CDRSC) statements to do this.

You must do this for all systems that are linked by NCF.

The application node and the cross-domain resource definitions are stored in the SYS1.VTAMLST data set, or in members of a data set that is in the same concatenation as SYS1.VTAMLST. For a detailed description of defining application program major nodes and cross-domain resources, see *VTAM Resource Definition Reference*.

The following example illustrates the definitions needed for a cross-domain setup between a controller and a tracker.

Note:

1. IBM Workload Scheduler for z/OS requires that the application name and the ACBNAME are the same.
2. IS1MVS1 and IS1MVS2 are only sample names.

At the *controller*:

1. Define the NCF controller application. Add a VTAM APPL statement like this to the application node definitions:

```
controller VTAM applications
VBUILD TYPE=APPL
  OPCCONTR APPL  VPACING=10,          C
                  ACBNAME=OPCCONTR
```

2. Define the NCF tracker application. Add a definition like this to the cross-domain resource definitions:

```
controller VTAM cross-domain resources
VBUILD TYPE=CDRSC
  OPCTRK1 CDRSC  CDRM=IS1MVS2
```

At the *tracker*:

1. Define the NCF tracker application. Add a VTAM APPL statement like this to the application node definitions:

```
tracker VTAM applications
VBUILD TYPE=APPL
  OPCTRK1 APPL  ACBNAME=OPCTRK1,    C
                  MODETAB=EQQLMTAB,  C
                  DLOGMOD=NCFSPPARM
```

2. Define the NCF controller application. Add a CDRSC statement like this to the cross-domain resource definitions:

```
tracker VTAM cross-domain resources
VBUILD TYPE=CDRSC
OPCCONTR CDRSC CDRM=IS1MVS1
```

IS1MVS1 and IS1MVS2 are the cross-domain resource managers for the controller and the tracker, respectively.

At the *Datastore*:

1. Define the NCF Datastore application. Add a VTAM APPL statement like this to the application node definitions:

```
Datastore VTAM applications
VBUILD TYPE=APPL
OPCDST1 APPL ACBNAME=OPCDST1,
MODETAB=EQQLMTAB,
DLOGMOD=NCFSPPARM
```

2. Define the NCF controller application. Add a CDRSC statement like this to the cross-domain resource definitions:

```
Datastore VTAM cross-domain resources
VBUILD TYPE=CDRSC
OPCCONTR CDRSC CDRM=IS1MVS1
```

Adding NCF session parameters

You can define the session parameters for NCF either by adding the sample EQQLMTAB logon-mode table or by using your own table. If you use the sample table, assemble and link-edit the EQQLMTAB table into the SYS1.VTAMLIB library concatenation for all trackers where an NCF transmitter application is defined.

Note that the APPL statement that defines an NCF application at a tracker must contain the logon-mode-table information in the MODETAB and DLOGMOD parameters.

The EQQLMTAB member of the SEQQSKL0 library contains this logon table definition plus the JCL necessary to assemble and link-edit the table:

```
EQQLMTAB
//LOGON JOB STATEMENT PARAMETERS
//ASM EXEC PGM=ASMA90,PARM='OBJ,NODECK'
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
// DD DSN=SYS1.SISTMAC1,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSLIN DD DSN=&LOADSET,UNIT=SYSDA,SPACE=(80,(250,50)),
// DISP=(,PASS)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EQQLMTAB MODETAB
MODEENT LOGMODE=NCFSPPARM,
FMPROF=X'04',
TSPROF=X'04',
PRIPROT=X'F3',
SECPROT=X'F3',
COMPROT=X'0000',
PSERVIC=X'000000000000000000000000',
RUSIZES=X'8787'
MODEEND
END
//LINK EXEC PGM=IEWL,PARM='XREF,LIST,LET,CALL'
//SYSPRINT DD SYSOUT=*
```

```
//SYSMOD DD DSN=SYS1.VTAMLIB(EQQLMTAB),DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,50))
//SYSLIN DD DSN=&LOADSET,DISP=(OLD,DELETE)
```

If you choose to provide session parameters in another table or entry, modify the APPL definitions for the transmitter applications accordingly. Note that NCF uses an LU-type 0 protocol with a recommended minimum RU-size of 500 bytes. Do *not* specify an RU-size smaller than 32 bytes. NCF does not modify the session parameter specified in the LOGMODE table entry in any way.

For a complete description of logon mode tables and the macros that define them, see *VTAM Customization*.

COS table

No class of service (COS) table entry is specified for EQQLMTAB in the sample. Specify a COS entry that is valid in your VTAM environment unless you intend to use the default provided by VTAM.

The routing you specify in the COS entry should be fast and reliable so that unnecessary delays are not introduced in the IBM Workload Scheduler for z/OS remote job-tracking function.

Activating network resources

The VTAM network must be active when the NCF application is started so that network resources are available for the NCF sessions. All participating NCF-application minor nodes must be activated before the NCF application is started by the tracker.

You activate VTAM resources by entering the VARY NET command or by specifying automatic activation in the VTAM network-definition procedure used during VTAM startup. You can activate NCF-application minor nodes and CDRSC minor nodes directly, using the VARY ACT command. You can also activate them indirectly by activating their major nodes. See *VTAM Operation* for further information.

Diagnostic data set

If you have not already allocated the EQQDUMP diagnostic data set for the tracker or controller, do this now. NCF writes debugging information to this diagnostic data set when internal error conditions are detected. When diagnostic information is logged, the information is normally accompanied by a user abend.

Note: Update the IBM Workload Scheduler for z/OS started-task procedure with DD name EQQDUMP if this DD name is not already defined.

Step 16. Using TCP/IP for communication

Include this task when installing a scheduler component that will use TCP/IP for communication.

To use the Transmission Control Protocol/Internet Protocol (TCP/IP) for communication among IBM Workload Scheduler for z/OS systems, do the following:

- Ensure that TCP/IP protocol is available and the relative started task is started on your z/OS configuration.

- Include the necessary initialization statement options for each product component.

Initialization statements used for TCP/IP

IBM Workload Scheduler for z/OS started tasks use the following initialization statements for connecting the scheduler started tasks through TCP/IP:

ROUTOPTS

To identify all the TCP/IP remote destinations for the controller or standby controller. A ROUTOPTS statement is required for each controller and standby controller.

TRROPTS

To identify the controller for a tracker. A TRROPTS statement is required for each tracker on a controlled system.

FLOPTS

To identify all the TCP/IP data store remote destinations for the controller.

DSOPTS

To identify the controller for a Data Store.

TCPOPTS

An optional statement to specify the TCP/IP options for the local component. To identify the remote partner, use one of the previous statements.

Step 17. Activating support for the API

Include this task when installing a controller, or standby controller that you want to communicate with through the IBM Workload Scheduler for z/OS API.

IBM Workload Scheduler for z/OS uses LU to LU communication to pass data between an ATP and a subsystem through the API. To use API requests GET, PUT, and DELETE, the LU that the ATP sends requests to (the target LU) must be owned by the controller. For CREATE requests, if the target LU is not owned by an IBM Workload Scheduler for z/OS address space where an event-writer task is started, the ATP must send requests so that the events are broadcast on the target z/OS system. *Tivoli Workload Automation: Developer's Guide: Driving Tivoli IBM Workload Scheduler for z/OS* and *IBM Workload Scheduler for z/OS: Customization and Tuning* describe when a request is broadcast.

To activate support for the API, perform these actions in the order shown:

1. Define VTAM resources.
2. Update APPC options.
3. Activate IBM Workload Scheduler for z/OS support for APPC.

If you are installing a standby controller, perform corresponding actions on the standby system.

You might need to refer to one or more of these publications:

- *VTAM Resource Definition Reference*
- *APPC Management*
- *z/OS Initialization and Tuning Reference*
- *Tivoli Workload Automation: Developer's Guide: Driving Tivoli IBM Workload Scheduler for z/OS*, which documents the API


```

* Logmode table entry for resources capable of acting as LU 6.2      *
* devices for PC target.                                           *
*-----*
LU62SYS1 MODEENT                                                    C
      LOGMODE=LU62SYS1,                                           C
      RUSIZES=X'8989',                                           C
      SRCVPAC=X'00',                                             C
      SSNDPAC=X'01'
*-----*
* Logmode table entry for resources capable of acting as LU 6.2      *
* devices for host target.                                           *
*-----*
APPCHOST MODEENT                                                    C
      LOGMODE=APPCHOST,                                           C
      RUSIZES=X'8F8F',                                           C
      SRCVPAC=X'00',                                             C
      SSNDPAC=X'01'
      MODEEND
      END

```

Defining cross-domain resources

If the IBM Workload Scheduler for z/OS TP and the partner TP are not running in the same VTAM domain, ensure that their respective LUs can communicate by defining cross-domain resources. In this example, LU name IS1MVS1 is used for the system where the controller is activated, and IS1MVS2 for the system that the partner TP is running on.

On the IBM Workload Scheduler for z/OS controlling system:

```

Partner LU cross-domain resources
VBUILD TYPE=CDRSC
  LUMVS2 CDRSC  CDRM=IS1MVS2

```

On the partner system:

```

&opc LU cross-domain resources
VBUILD TYPE=CDRSC
  LU0PC CDRSC  CDRM=IS1MVS1

```

Updating APPC options

You must update APPC options to associate the IBM Workload Scheduler for z/OS scheduler (the subsystem) with the local LU that you defined earlier. Do this by updating the APPCPMnn member of SYS1.PARMLIB. Here is an example of an APPCPMnn member:

```

APPCPMnn example
LUADD                /* Add local LU to APPC config. */
  ACBNAME(IS4MEOP4)  /* Name of LU */
  SCHED(EOP4)        /* Scheduler name/OPC subsys name */
  TPDATA(SYS1.APPCTP) /* Profile data set for this LU */
  TPLEVEL(SYSTEM)    /* TP level for which LU searches */

```

The scheduler name must be the same as the IBM Workload Scheduler for z/OS subsystem name. In this example, the subsystem name is EOP4. A side information file is not used by IBM Workload Scheduler for z/OS. However, the LU must be associated with a TP profile data set; you need not specify a profile for IBM Workload Scheduler for z/OS in the data set because IBM Workload Scheduler for z/OS does not use TP profiles.

If you must allocate a TP profile data set, you can run a job such as:

```

//ALTPDSET JOB STATEMENT PARAMETERS
//TPSAMPLE EXEC PGM=IDCAMS
//VOLOUT DD DISP=OLD,UNIT=3380,VOL=SER=volser

```

```

//SYSPRINT DD  SYSOUT=*
//SYSIN      DD  *
      DEFINE CLUSTER (NAME(SYS1.APPCTP) -
        VOLUMES(volser) -
        INDEXED REUSE -
        SHAREOPTIONS(3 3) -
        RECORDSIZE(3824 7024) -
        KEYS(112 0) -
        RECORDS(300 150)) -
      DATA -
        (NAME(SYS1.APPCTP.DATA)) -
      INDEX -
        (NAME(SYS1.APPCTP.INDEX))

```

TP profile data sets are VSAM KSDS data sets.

Activating support for APPC

When you have defined the necessary VTAM resources and updated APPC options, you can activate IBM Workload Scheduler for z/OS support for APPC. Do this by specifying APPCTASK(YES) on the OPCOPTS statement. Perform this action when you have completed all other actions and before you start to use the IBM Workload Scheduler for z/OS API.

Step 18. Activating support for the product dialog and programming interface using the server

Include this task when activating an IBM Workload Scheduler for z/OS server. To use the Dynamic Workload Console, see "Step 21. Activating support for Dynamic Workload Console" on page 155.

The IBM Workload Scheduler for z/OS dialogs and programming interface can be used on a z/OS system other than the system where the controller is running. A server is required, running on the same z/OS system as the controller.

The dialogs and the programming interface on the remote z/OS system communicate with the server using APPC or TCP/IP. The EQQXLUSL help panels flow describes how to activate the communication with the server using the dialog.

The APPC communication requires also the VTAM and APPC definitions that are described in the following sections.

For further information, see the following publications:

VTAM Resource Definition Reference
APPC Management
MVS Initialization and Tuning Reference

See also member EQQVTAMS in the EQQJOBS output library or SEQQSAMP library.

To activate the APPC communication, perform the following steps:

1. The server tasks run on the same system as the controller.
2. On the system where the servers and the controller run, you must define the following LUs:
 - One LU for the server 'without' the BASE keyword, and specifying the server started task as SCHEDULER.

3. On the system where you want to enable TSO users to communicate with IBM Workload Scheduler for z/OS via the server interface, you must define one LU with the keywords BASE and SCHED; that is:
 - An APPC MASTER LU 'with' the BASE keyword, and specifying SCHED(ASCH). This LU is not for IBM Workload Scheduler for z/OS; it is an APPC requirement that there be a BASE LU with SCHED(ASCH) on every system where APPC is used.
4. When the servers and the APPC address space are all started, you will see messages in the SYSLOG and server EQQMLOGs, stating that communication has been established between the server and APPC. These messages are displayed during the first start and after an IPL.
5. The dialog user then selects option 0.1 and specifies the name of the controller subsystem with which he wants to communicate, and the LUNAME of the server via which that communication is to be routed. For more information on specifying these values, press the PF1 (help) on panel EQQXLUSL.

The IBM Workload Scheduler for z/OS dialog code in the TSO logon address space then sends an APPC request which is picked up by the APPC BASE LU on that system and routed to the (server) LU specified in the request. The server then passes the dialog data to the controller across the z/OS subsystem interface, serving as a local proxy for the dialog user. The controller cannot tell whether it is talking to a local ISPF dialog user, or to a remote user via a server.

Defining VTAM resources for the product dialog and program interface using the server

If you intend to use the IBM Workload Scheduler for z/OS programming interface or dialog from a remote system, you need to activate the APPC support on the remote system.

Assure that there is a LU defined as default LU for the APPC communication (BASE LU) in the APPCPMnn parmlib member. If none is defined, add it as follows:

The IBM Workload Scheduler for z/OS dialog and programming interface use the default APPC support defined on the system on which the functions are used. To activate this support:

1. Define a default VTAM APPL supporting APPC:

```

VBUILD TYPE=APPL
APPCOUT APPL APPC=YES
        ACBNAME=APPCOUT
        ...

```

2. Update the APPCPMnn member of SYS1.PARMLIB for the default VTAM APPL defined above:

```

LUADD                                /* Add local LU to APPC config. */
ACBNAME(APPCOUT) /* Name of LU */
SCHED(ASCH)      /* No scheduler associated */
BASE             /* default LU for the system */
TPDATA(SYS1.APPCTP) /* Profile data set for this LU */
TPLEVEL(SYSTEM)  /* TP level for which LU search */

```

3. Add any cross-domain resource definition needed to resolve VTAM addressing.

Defining VTAM resources for the server

Start by defining the associated VTAM resources.

Defining a local LU for the server

Define a local LU in a member in the SYS1.VTAMLST concatenation on the system where you are installing IBM Workload Scheduler for z/OS. This example shows how a VTAM APPL statement might be defined:

Local LU for the server definition

```
VBUILD TYPE=APPL
  IS4MEOP5 APPL APPC=YES,           C
              AUTOSES=5,           C
              DMINWNL=3,           C
              DMINWNR=6,           C
              DSESLIM=20,          C
              MODETAB=APPCMODE,    C
              SECACPT=ALREADYV,    C
              SRBEXIT=YES,         C
              VERIFY=OPTIONAL,     C
              VPACING=2
```

The LU is called IS4MEOP5 and uses the logon-mode table APPCMODE.

The maximum number of TSO dialog users and PIF programs that can simultaneously access an IBM Workload Scheduler for z/OS controller via a single server depends on the DSESLIM parameter of the VTAM LU for that server. Once the specified number of sessions has been established, all subsequent users and PIF programs that try to use that server will hang until one of the existing sessions ends.

The number of servers required by an installation depends on how extensive PIF applications are used. While it can be sufficient with one server for the dialogs, a number of servers can be required for the PIF applications. PIF applications that are frequently used and with long execution time might need separate servers.

Defining logon modes for the server

The logon-mode table, which you specify in the LU APPL definition statement, must be in the SYS1.VTAMLIB concatenation.

The server support requires logon-mode table entries as specified in the following uncompiled example:

```
APPCDIA logon-mode table for server
*-----*
* Logmode table entry for the dialogs and the programming interface *
*-----*
APPCDIA MODEENT           C
      LOGMODE=APPCDIA,    C
      RUSIZE=X'8888',     C
      SRCVPAC=X'00',     C
      SSNDPAC=X'01',     C
      MODEENT             C
APPCFIF MODEENT           C
      LOGMODE=APPCFIF,    C
      RUSIZE=X'8888',     C
      SRCVPAC=X'00',     C
      SSNDPAC=X'01',     C
      MODEENT             C
```

The RUSIZE gives a user size for sending buffer of 2048 bytes and a receiving buffer of 4096 bytes.

Updating APPC options for the server

You must update APPC options to associate the server (and controller) with the scheduler that you defined earlier. Do this by updating the LUADD statement in the APPCPMnn member of SYS1.PARMLIB. Here is an example of an APPCPMnn member:

```
APPCPMnn example
LUADD                /* Add local LU to APPC config. */
  ACBNAME(IS4MEOP5)  /* Name of LU */
  SCHED(EOP5)        /* Scheduler name/OPC subsys name */
  TPDATA(SYS1.APPCTP) /* Profile data set for this LU */
  TPLEVEL(SYSTEM)    /* TP level for which LU searches */
```

The scheduler name in LUADD must be the same as the scheduler name of the scheduler server. In this example it is EOP5.

Each server identifies itself to APPC as an APPC scheduler with the same name as the started task name. If the SCHEDULER keyword in the SERVOPTS statement is specified, this name is used instead of the started task name.

Defining VTAM resources in a parallel sysplex

In an installation with a Parallel Sysplex[®] where the scheduler can start on any of a number of z/OS images, each z/OS image within the parallel sysplex should have the same local LU name for a given server. The same LU name must not exist in any other network interconnected to the parallel sysplex network; identical LU names within network, unique LU names across networks.

For details on the parallel sysplex installation, see “Step 14. Using XCF for communication” on page 141.

The LU name (the APPL statement name) should be given with a wildcard character, in case the scheduler works in a parallel sysplex and is not set up to run on a specific z/OS image. The APPL statement will then become a Model Application Program Definition, for the identically named LUs on the z/OS images where the scheduler might start. The wildcard character should be chosen such that one model definition is set up for the controller and one model definition for each of the servers. The optional ACBNAME parameter must be omitted, the name of the APPL statement is then used as the ACBNAME.

For example, say the scheduler can start on z/OS images MVS1 and MVS2 in a parallel sysplex. The LU name for controller OPCB is IS4MOPCB, and there are three servers to handle the communication to OPCB, OPCBCOM1, OPCBCOM2 and OPCBCOM3, with LU names IS4MSV1B, IS4MSV2B and IS4MSV3B. VTAM Version 4 Release 3 is available. The following model definitions could then be used (a '?' in the APPL statement name represents a single unspecified character):

```
IS4MOP?B  APPL APPC=YES,...
IS4MS?1B  APPL APPC=YES,...
IS4MS?2B  APPL APPC=YES,...
IS4MS?3B  APPL APPC=YES,...
```

Note that the wildcard character must be chosen such that the no other VTAM LU name than the intended LU name matches the model definition.

Starting the server

You can start the server by using the z/OS START command, or you can have the controller start and stop the server automatically. In the latter case, include the servers (srv1, srv2, ...) on the OPCOPTS statement in the IBM Workload Scheduler for z/OS parameter library.

A SERVOPTS statement is required in the parameters file. All SERVOPTS keywords can be left out and defaulted.

Step 19. Activating support for the end-to-end scheduling with fault tolerance capabilities

To schedule jobs on IBM Workload Scheduler distributed fault-tolerant agents, activate the end-to-end scheduling with fault tolerance capabilities. Follow these steps:

1. Run EQQJOBS and specify Y for the END-TO-END WITH FAULT TOLERANCE feature.
2. Allocate the data set running the generated EQQPCS06 sample.
3. Create and customize the work directory by running the generated EQQPCS05 sample.
4. Define CPU configuration and domain organization by using the CPUREC and DOMREC statements in a PARMLIB member (the default member name is TPLGINFO).
5. Define Windows user IDs and passwords by using the USRREC statement in a PARMLIB member (the default member name is USRINFO). To encrypt the passwords, run the EQQE2EPW JCL contained in the sample EQQBENCR JCL generated by EQQJOBS.

If you do not want to set the password through the USRREC statement (either in plaintext or encrypted), define the user and password locally on the Windows workstation by using the users utility, and set LOCALPSW=YES in the TOPOLOGY statement. For detailed information about the users script, see the *IBM Workload Scheduler for z/OS: End-to-end Scheduling with Fault Tolerance Capabilities* manual.

6. Define the end-to-end configuration by using the TOPOLOGY statement in a PARMLIB member (the default member name is TPLGPARM). In this statement, specify the following:
 - For the TPLGYMEM keyword, write the name of the member used in step 4.
 - For the USRMEM keyword, write the name of the member used in step 5.
7. Add the TPLGYSRV keyword to the OPCOPTS statement to specify the server name that will be used for end-to-end communication.
8. Add the TPLGYPRM keyword to the SERVOPTS statement to specify the member name used in step 6. This step activates end-to-end communication in the Server.
9. Add the TPLGYPRM keyword to the BATCHOPT statement to specify the member name used in step 6. This step activates the end-to-end scheduling with fault tolerance capabilities feature in the Daily Planning batch programs.

Activating server support for the end-to-end scheduling with fault tolerance capabilities

Customize the INIT and SERVOPTS initialization parameters to set up the correct server environment. For example:

```
SERVOPTS SUBSYS (OPCX)
          PROTOCOL (E2E)
          TPLGYPRM(TPLGY)
```

For more information about these statements, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

You can start the server by using the z/OS START command, or you can have the controller start and stop the server automatically. In the latter case, include the server (srv1) in the OPCOPTS statement in the IBM Workload Scheduler for z/OS parameter library. The server with TCP/IP support requires access to the C language runtime library (either as STEPLIB or as LINKLIST). If you have multiple TCP/IP stacks, or a TCP/IP started task with a name different from 'TCPIP', then use the TCPIPJOBNAME parameter of the TOPOLOGY statement.

You always have to define OMVS segments for server started tasks.

Step 20. Activating support for end-to-end scheduling with z-centric capabilities

To schedule jobs on IBM Workload Scheduler distributed z-centric agents, activate the end-to-end scheduling with z-centric capabilities. Follow these steps:

1. Define the z-centric agent destinations in the ROUTOPTS initialization statement.
2. Customize the connection parameters in the HTTPOPTS initialization statement.

Note: Use this statement to activate or disable the SSL connection protocol . If you want to disable the SSL connection, you can either:

- Specify neither SSLKEYRING nor SSLPORT keywords.
 - Specify SSLPORT(0).
3. Optionally, activate the output collector started task to retrieve job logs and copy them to a JES SYSOUT for processing by an external output management product. Customize the HTTPOPTS (OUTPUTCOLLECTOR and JLOGHDRTEMPL keywords), OPCOPTS (OUTCOL keyword), and OUCOPTS initialization statements.

For details about the configuration steps, see *Scheduling End-to-end with z-centric Capabilities*.

Step 21. Activating support for Dynamic Workload Console

Perform this step if you want to use the Dynamic Workload Console to design and run your workload.

Prerequisites

Before using the Dynamic Workload Console, you need to install the console and the IBM Workload Scheduler for z/OS connector.

The z/OS connector forms the bridge between the console and IBM Workload Scheduler for z/OS.

The Dynamic Workload Console installation process installs also the z/OS connector. During the Dynamic Workload Console installation, you can define a

z/OS engine to connect to one IBM Workload Scheduler for z/OS. To add more z/OS engines definition after the installation process, see “Managing z/OS engines by using WebSphere Application Server tools” on page 219.

The console communicates with the product through the z/OS connector and the scheduler server by using the TCP/IP protocol. The console needs the server to run as a started task in a separate address space. The server communicates with IBM Workload Scheduler for z/OS and passes the data and return codes back to the z/OS connector.

Perform the following task:

- Install and configure the Dynamic Workload Console as described in Part 4, “Dynamic Workload Console,” on page 247.

Considerations

The security model implemented for the Dynamic Workload Console is similar to that already implemented by other Tivoli products that have been ported to z/OS (namely Tivoli User Administration and Tivoli Security Management).

All versions of the Dynamic Workload Console use WebSphere Application Server to handle the initial user verification. In all cases, however, it is necessary to obtain a valid corresponding RACF user ID to be able to work with the security environment in z/OS.

Note: You cannot control the port from which the Dynamic Workload Console server started task replies to a request from the z/OS connector. The response ports are randomly selected. Therefore, if there is a firewall between the Dynamic Workload Console server and the z/OS Connector, that firewall must permit outgoing traffic from all ports above 1023.

To optimize the thread handling between z/OS connector and the scheduler server, you can group console users by RACF user ID. To define this grouping, associate a list of console users to the same RACF user ID, by editing the following file:

```
<JazzSM_INSTALL_DIR>\profile\properties\TWSZ0SConnConfig.properties
```

Where <JazzSM_INSTALL_DIR> is the Jazz for Service Management extension for WebSphere installation directory. For more information about the Jazz for Service Management extension for WebSphere prerequisite for the Dynamic Workload Console see Chapter 12, “Dynamic Workload Console prerequisites,” on page 255.

In TWSZ0SConnConfig.properties set the last two properties as follows:

```
com.ibm.tws.zconn.usr.mapping.enable=true  
com.ibm.tws.zconn.usr.mapping.file=mapping_file_path\mapping_file
```

where *mapping_file* is the name of the file that contains the mapping between console user and RACF user ID, as in the following example:

```
engine=zos1919 user=twuser1,twuser2 zosuser=zos1919user1  
user=twuser3,twuser4 zosuser=zos1919user2
```

Note: Every time you update the Dynamic Workload Console, the TWSZ0SConnConfig.properties file is replaced and any customized property you might have set is lost. To prevent this from occurring save your current version of the file and, after updating the Dynamic Workload Console, set again the properties you had customized.

Activating server support for the Dynamic Workload Console

Customize the INIT and SERVOPTS initialization parameters to set up the correct server environment. For example:

```
SERVOPTS SUBSYS (OPCX)
          USERMAP (USERS)
          PROTOCOL (TCP)
          PORTNUMBER (425)
          CODEPAGE (IBM-037)
INIT      CALENDAR (DEFAULT)
```

For more information, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

You can start the server by using the z/OS START command, or you can have the controller start and stop the server automatically. In the latter case, include the servers (srv1, srv2, ...) on the OPCOPTS statement in the IBM Workload Scheduler for z/OS parameter library. The server with TCP/IP support requires access to the C language runtime library (either as STEPLIB or as LINKLIST). If you have multiple TCP/IP stacks, or a TCP/IP started task with a name different from 'TCPIP', then a SYSTCPD DD card is required pointing to a TCP/IP data set containing the TCPIPJOBNAME parameter.

You always have to define OMVS segments for IBM Workload Scheduler for z/OS server started tasks.

Step 22. Activating support for the Java utilities

This section describes actions that are required if you want to use one of the following features:

- Dynamic Workload Console reporting.
- Event-driven workload automation for data set triggering, with centralized deploy process.

For details about these features, see *IBM Workload Scheduler for z/OS: Managing the Workload*.

As installation actions, perform the following steps:

1. Install IBM Java SDK for z/OS platforms. For information about how to install it, see *IBM SDK for z/OS platforms, Java Technology Edition*.
2. Copy the JZOS Java Launcher load module (JVMLDM66) from the JAVA_HOME directory to the SYS1.SIEALNKE system data set. For details about customizing the JZOS Java Launcher, see *JZOS Batch Launcher and Toolkit function in IBM SDK for z/OS*.
3. Make sure that you applied FMID JWSZ933.
4. Run EQQJOBS with the option to enable JAVA utilities, to create the EQQPCS08 sample JCL.
5. Customize EQQPCS08 and submit it.
6. Define the TRGOPT initialization statement in a member of the EQQPARM library.
7. Define the event rule in XML format. You can use a partitioned data set member to be used as input for the following step. The SEQQSAMP library contains EQQXML01 member as sample of event rule definition.

8. Select option 1.7.3 from the main menu, edit and submit the job to produce the configuration files.

Step 23. Activating support for FIPS standard over SSL secured connections

Secure Sockets Layer (SSL) is a communications protocol that provides secure communications over an open communications network (for example, the Internet).

Federal Information Processing Standard Security Requirements for Cryptographic Modules, referred to as FIPS 140-2, is a standard published by the National Institute of Standards and Technology (NIST). Organizations can require compliance to the FIPS 140-2 standard to provide protection for sensitive or valuable data to cryptographic-based security systems.

System SSL was designed to meet the Federal Information Processing Standard - FIPS 140-2 Level 1 criteria.

System SSL can run in either "FIPS mode" or "non-FIPS mode". By default, System SSL runs in "non-FIPS" mode.

IBM Workload Scheduler for z/OS uses the System SSL configuration. To run IBM Workload Scheduler for z/OS in "FIPS mode", you must enable FIPS compliance over System SSL connections.

For more information about:

- How to enable FIPS compliance over System SSL connections
- System prerequisites
- Differences between FIPS mode and non-FIPS mode algorithm support and keys sizes

see *IBM z/OS Cryptographic Services System Secure Sockets Layer Programming manual, Chapter 4. System SSL and FIPS 140-2* .

Note: Algorithm support and key sizes are different when FIPS-mode is set.

The IBM Workload Scheduler for z/OS communications that can implement FIPS 140-2 Level 1 standards over a secured SSL connection are:

Backup Controller Communication task for communication between the controller and backup controller

To enable FIPS 140-2 compliance for this communication, set ENABLEFIPS to YES in the BKPTOPTS initialization statement.

For more information about the BKPTOPTS initialization statement, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

HTTP client and server and output collector for communication with the z-centric agents

For information about how to enable FIPS 140-2 compliance over IBM Workload Scheduler for z/OS server SSL secured connection, see the *IBM Workload Scheduler for z/OS; Scheduling End-to-end with z-centric Capabilities* manual.

Fault Tolerant end-to-end task for communication with distributed agents

For information about how to enable up FIPS 140-2 compliance over IBM Workload Scheduler for z/OS server SSL secured connection, see the *IBM Workload Scheduler for z/OS: Scheduling End-to-end with Fault Tolerance Capabilities* manual.

IP task for communication between the controller and tracker, server, datastore, remote ISPF dialog

To enable FIPS 140-2 compliance for this communication, set ENABLEFIPS to YES in the TCPOPTS initialization statement.

For more information about the TCPOPTS initialization statement, see *IBM Workload Scheduler for z/OS: Customization and Tuning*.

Note: Ensure that you run the initialization statements before starting IBM Workload Scheduler for z/OS to ensure the use of FIPS standards over that specific SSL secured connection.

You do not need to apply FIPS compliance to all communications; you can decide which communications run in "FIPS-mode" and which run in "non-FIPS mode".

If FIPS compliance is not required by your organization, you can continue to use SSL for secure connections across your network.

Chapter 5. Verifying your installation

What to consider when verifying your installation.

Perform this task for a tracker controller or standby controller.

Use the following procedures to verify your installation of a single IBM Workload Scheduler for z/OS address space, or your configuration.

Overview of verification

When you have installed a tracker, controller, standby controller, or server, start it and perform initial verification procedures. To fully verify IBM Workload Scheduler for z/OS, start all the address spaces in your configuration, and create database entries, a long-term plan, and a current plan. This is required to verify job submission and connections between systems, and requires some knowledge of the product. Therefore, verification is divided into two parts:

- Initial verification of individual IBM Workload Scheduler for z/OS address spaces.
- Verification of your configuration.

You can therefore perform some verification tasks without needing to know more detailed aspects of IBM Workload Scheduler for z/OS. When you are more familiar with the product components and functions, you can perform further testing.

The following topics are described:

- “Verifying installation of a tracker”
- “Verifying installation of a controller and dialogs” on page 167
- “Verifying installation of a standby controller” on page 170
- “Verifying installation of the Restart and Cleanup function” on page 172
- “Verifying configuration” on page 174

If you are installing a tracker and controller in the same address space, review the initial verification procedures for both a tracker and a controller.

Verifying installation of a tracker

When you have completed the installation tasks for a tracker, perform initial verification of the tracker. Because connections and the submission of work cannot be verified in isolation, you can perform further verification of the tracker when you have installed the controlling system, established connections between IBM Workload Scheduler for z/OS systems, and created a current plan. These verification tasks are described in “Verifying configuration” on page 174.

To initially verify the tracker, perform these tasks:

1. Follow the appropriate procedures for the IBM Workload Scheduler for z/OS subsystem that you are installing.
2. Ensure that you have completed all the necessary installation tasks.
3. Start the tracker and check the message log (EQQMLOG).
4. Verify that tracking events are created in the event data set (EQQEVDS).

5. Perform problem determination for tracking events if events are missing from the event data set.

For TCP/IP connections only, ensure that a valid current plan exists before verifying the tracker.

Ensuring that all installation tasks are complete

Ensure that you have performed all the installation tasks that are needed for your IBM Workload Scheduler for z/OS service. That is, you should have:

- Followed the appropriate procedures for the IBM Workload Scheduler for z/OS subsystem that you are installing
- Installed the required JES and SMF exits, and verified that they are active
- Created a JCL procedure for the tracker
- Allocated required data sets
- Given the security access for the subsystem to access the data sets
- Specified the initialization statements in the parameter library (EQQPARM)
- Included the tracker in the same XCF group as the controller, if the tracker uses an XCF connection
- Defined a VTAM LU name for the tracker and activated the VTAM resources, if the tracker uses an NCF connection.

Checking the message log (EQQMLOG)

Start the tracker.

When the tracker is started, check the message log:

- Check that the return code for all initialization options is 0 (message EQQZ016I).
- Ensure that all required subtasks are active.

– The data-router and submit tasks are always started. You should see these messages:

```
EQQZ005I OPC SUBTASK DATA ROUTER TASK IS BEING STARTED
EQQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE
```

```
EQQZ005I OPC SUBTASK JOB SUBMIT TASK IS BEING STARTED
EQQS001I THE SUBMIT TASK HAS STARTED
```

– Also, verify that the tracker has started an event writer. You should see these messages:

```
EQQZ005I OPC SUBTASK EVENT WRITER IS BEING STARTED
EQQW065I EVENT WRITER STARTED
```

- Examine error messages.

Note: The first time the event writer is started, it formats the event data set. Ignore the SD37 abend code that is issued during the formatting process.

If you see error messages in the message log for an event reader or an NCF connection, this is because you cannot fully verify an event-reader function or NCF connection until the controller is active and a current plan exists. Active tracker-connection messages for XCF connections are written to the controller message log when the controller is started. If you have specified any of these functions, follow the procedures in “Verifying configuration” on page 174 when you have completed initial verification procedures.

- Check that your log is complete.

“Sample message log for a tracker” on page 181 shows an example of the MLOG for a tracker. If your log seems to be incomplete, information might be in a buffer. If you are unsure whether the log is complete, issue a dummy modify command like this: F sname,xx. Message EQQZ049E is written to the log when the command is processed. This message will be the last entry in the log.

Verifying tracking events

The next verification phase is to check that the tracker is collecting tracking-event information and writing it to the event data set (EQQEVDS).

IBM Workload Scheduler for z/OS job tracking works correctly only if it receives information about status changes for all jobs or started tasks to be tracked. Job tracking gets this information from SMF and JES exits. These exits gather the necessary information, and an exit record is added to the IBM Workload Scheduler for z/OS event-writer queue via ECSA buffers.

The event writer

The event writer removes the event from its queue and creates an event record that is written to an event data set. The event writer also forwards the event if it has been started with an event-reader function. If a separate event reader is used, the event is read from the event data set. In either case, the reader task uses the connection with the controller to transfer the event to a queue at the controller. The event-manager subtask then processes the event and the current plan is updated.

The event data set

The event data set is needed to even out any difference in the rate that events are being generated and processed, and to prevent events from being lost if the IBM Workload Scheduler for z/OS address space or a subtask must be restarted. The first byte in an exit record is A if the event is created on a JES2 system, or B if the event is created on a JES3 system. This byte is found in position 21 of a standard event record, or position 47 of a continuation (type N) event. Bytes 2 and 3 in the exit record define the event type. These event types are generated by IBM Workload Scheduler for z/OS for jobs and started tasks:

- 1 Reader event. A job has entered the JES system.
- 2 Job-start event. A job has started to execute.
- 3S Step-end event. A job step has finished executing.
- 3J Job-end event. A job has finished executing.
- 3P Job-termination event. A job has been added to the JES output queues.
- 4 Print event. An output group has been printed.
- 5 Purge event. All output for a job has been purged from the JES system.

If any of these event types are not being created in the event data set (EQQEVDS), a problem must be corrected before IBM Workload Scheduler for z/OS is started in production mode.

Note:

1. The creation of step-end events (3S) depends on the value you specify in the STEPEVENTS keyword of the EWTROPTS statement. The default is to create a step-end event only for abending steps in a job or started task.
2. The creation of print events depends on the value you specify in the PRINTEVENTS keyword of the EWTROPTS statement. By default, print events are created.

Perform these actions to verify that events are being created on your system:

1. Run a job:

- a. Submit a job like the following, ensuring that the output is written to a non-held output class:

```
Test job
//VERIFY1 JOB STATEMENT PARAMETERS

//VERIFY EXEC PGM=IEBGENER
//*
//SYSPRINT DD DUMMY
//SYSUT2 DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD *
        SAMPLE TEST OUTPUT STATEMENT 1
/*
```

- b. Verify that the job has executed, printed, and purged.
- c. Browse the EQQEVDS data set using the ISPF/PDF browse facility. You should find these events on the event data set:
- Type 1 event
 - Type 2 event
 - Type 3J event
 - Type 3P event
 - Type 4 event
 - Type 5 event.

The events are prefixed with A for JES2 or B for JES3. You might also find type 3S as events, depending on the value specified on the STEPEVENTS keyword of the EWTROPTS initialization statement.

2. Repeat step 1 for a started task.

Performing problem determination for tracking events

Problem determination depends on which event is missing and whether the events are created on a JES2 or JES3 system. In Table 33, the first column refers to the event type that is missing, and the second column tells you what action to perform. Events created on a JES2 system are prefixed with A, and events created on a JES3 system with B. The first entry in the table applies when all event types are missing (when the event data set does not contain any tracking events).

Table 33. Problem determination for missing tracking events

Type	Problem determination actions
All	<ol style="list-style-type: none"> 1. Verify in the EQQMLOG data set that the event writer has started successfully. 2. Verify that the definition of the EQQEVDS ddname in the IBM Workload Scheduler for z/OS started-task procedure is correct (that is, events are written to the correct data set). 3. Verify that the required exits have been installed. 4. Verify that the IEFSSNmm member of SYS1.PARMLIB has been updated correctly, and that an IPL of the z/OS system has been performed since the update.

Table 33. Problem determination for missing tracking events (continued)

Type	Problem determination actions
A1	<p>If both A3P and A5 events are also missing:</p> <ol style="list-style-type: none"> 1. Verify that the IBM Workload Scheduler for z/OS version of the JES2 exits 7 and 51 routines have been correctly installed. Use the JES commands \$T EXIT(7) and \$T EXIT(51) or \$DMODULE(OPCAXIT7) and \$DMODULE(TWSXIT51). 2. Verify that the JES2 initialization data set contains a LOAD statement and an EXIT7 statement for the IBM Workload Scheduler for z/OS version of JES2 exit 7 (OPCAXIT7). For z/OS version 1.7 or later, verify also that the JES2 initialization data set contains a LOAD statement and an EXIT51 statement for the version of JES2 exit 51 (TWSXIT51) 3. Verify that the exit has been added to a load module library reachable by JES2 and that JES2 has been restarted since this was done. <p>If either A3P or A5 events are present in the event data set, call an IBM service representative for programming assistance.</p>
B1	<ol style="list-style-type: none"> 1. Verify that the IBM Workload Scheduler for z/OS version of the JES3 exit IATUX29 routine has been correctly installed. 2. Verify that the exit has been added to a load-module library that JES3 can access. 3. Verify that JES3 has been restarted.
A2/B2	<ol style="list-style-type: none"> 1. Verify that the job for which no type 2 event was created has started to execute. A type 2 event will not be created for a job that is flushed from the system because of JCL errors. 2. Verify that the IEFUJI exit has been correctly installed: <ol style="list-style-type: none"> a. Verify that the SMF parameter member SMFPRM$_{nn}$ in the SYS1.PARMLIB data set specifies that the IEFUJI exit should be called. b. Verify that the IEFUJI exit has not been disabled by an operator command. c. Verify that the correct version of IEFUJI is active. If SYS1.PARMLIB defines LPALIB as a concatenation of several libraries, z/OS uses the first IEFUJI module found. d. Verify that the library containing this module was updated by the IBM Workload Scheduler for z/OS version of IEFUJI and that z/OS has been IPLed since the change was made.

Table 33. Problem determination for missing tracking events (continued)

Type	Problem determination actions
A3S/B3S	<p>If type 3J events are also missing:</p> <ol style="list-style-type: none"> 1. Verify that the IEFACRT exit has been correctly installed. 2. Verify that the SMF parameter member SMFPRMmm in the SYS1.PARMLIB data set specifies that the IEFACRT exit should be called. 3. Verify that the IEFACRT exit has not been disabled by an operator command. 4. Verify that the correct version of IEFACRT is active. If SYS1.PARMLIB defines LPALIB as a concatenation of several libraries, z/OS uses the first IEFACRT module found. 5. Verify that this library was updated by the IBM Workload Scheduler for z/OS version of IEFACRT and that z/OS has been IPLed since the change was made. <p>If type 3J events are not missing, verify, in the EQQMLOG data set, that the event writer has been requested to generate step-end events. Step-end events are created only if the EWTROPTS statement specifies STEPEVENTS(ALL) or STEPEVENTS(NZERO) or if the job step abended.</p>
A3J/B3J	<p>If type 3S events are also missing, follow the procedures described for type 3S events.</p> <p>If type 3S events are not missing, call an IBM service representative for programming assistance.</p>
A3P	<p>If A1 events are also missing, follow the procedures described for A1 events.</p> <p>If A1 events are not missing, call an IBM service representative for programming assistance.</p>
B3P	<ol style="list-style-type: none"> 1. Verify that the IBM Workload Scheduler for z/OS version of the JES3 exit IATUX19 routine has been correctly installed. 2. Verify that the exit has been added to a load-module library that JES3 can access. 3. Verify that JES3 has been restarted.
A4/B4	<ol style="list-style-type: none"> 1. If you have specified PRINTEVENTS(NO) on the EWTROPTS initialization statement, no type 4 events are created. 2. Verify that JES has printed the job for which no type 4 event was created. Type 4 events will not be created for a job that creates only held SYSOUT data sets. 3. Verify that the IEFU83 exit has been correctly installed: <ol style="list-style-type: none"> a. Verify that the SMF parameter member SMFPRMmm in the SYS1.PARMLIB data set specifies that the IEFU83 exit should be called. b. Verify that the IEFU83 exit has not been disabled by an operator command. c. Verify that the correct version of IEFU83 is active. If SYS1.PARMLIB defines LPALIB as a concatenation of several libraries, z/OS uses the first IEFU83 module found. d. Verify that the library containing this module was updated by the IBM Workload Scheduler for z/OS version of IEFU83 and that z/OS has been IPLed since the change was made. e. For JES2 users (A4 event), ensure that you have not specified TYPE6=NO on the JOBCLASS and STCCLASS statements of the JES2 initialization parameters.

Table 33. Problem determination for missing tracking events (continued)

Type	Problem determination actions
A5	<ol style="list-style-type: none"> 1. Verify that JES2 has purged the job for which no A5 event was created. 2. Ensure that you have not specified TYPE26=NO on the JOBCLASS and STCCCLASS statements of the JES2 initialization parameters. 3. If A1 events are also missing, follow the procedures described for A1 events. 4. If A1 events are not missing, call an IBM service representative for programming assistance.
B5	<ol style="list-style-type: none"> 1. Verify that JES3 has purged the job for which no B5 event was created. 2. If B4 events are also missing, follow the procedures described for B4 events. 3. If B4 events are not missing, call an IBM service representative for programming assistance.

Verifying installation of a controller and dialogs

When you have completed the installation tasks for a controller, perform initial verification of the controller. Because connections and the submission of work cannot be verified in isolation, you can perform further verification of the controller when you have installed the controlling system, established connections between IBM Workload Scheduler for z/OS systems, and created a current plan. “Verifying configuration” on page 174 describes these verification tasks.

To initially verify the controller, perform the following steps:

1. Ensure that you have completed the installation tasks.
2. Start the controller, and check the message log (EQQMLOG).
3. Check that you can access IBM Workload Scheduler for z/OS data via the dialogs, and that authority checking is functioning as required.

If you encounter an error during verification, see “Performing problem determination” on page 169.

Ensuring that all installation tasks are complete

Check that you have:

- Created a started-task procedure for the controller
- Allocated data sets
- Given security authority to the started task to access its data sets
- Specified the initialization statements in the parameter library (EQQPARM)
- Included the controller in an XCF group, if it uses an XCF connection
- Defined a VTAM application ID for the controller and activated the VTAM resources, if it uses an NCF connection
- Updated SYS1.PARMLIB and defined VTAM resources, if users communicate with the controller through the IBM Workload Scheduler for z/OS API or if the IBM Workload Scheduler for z/OS server is used.
- Set up the ISPF environment for IBM Workload Scheduler for z/OS dialog users.

Checking the message log (EQQMLOG)

Start the controller. See *IBM Workload Scheduler for z/OS: Managing the Workload* for information on starting and stopping IBM Workload Scheduler for z/OS.

When the controller is started, check the message log:

- Ensure that the return code for all initialization options is 0 (message EQQZ016I).
- Check that all required subtasks are active.

Look for these messages when the controller is started:

Active general-service messages

```
EQQZ005I OPC SUBTASK GENERAL SERVICE IS BEING STARTED
EQQZ085I OPC SUBTASK GS EXECUTOR 01 IS BEING STARTED
EQQG001I SUBTASK GS EXECUTOR 01 HAS STARTED
⋮
EQQG001I SUBTASK GENERAL SERVICE HAS STARTED
```

Note: The preceding messages, EQQZ085I and EQQG001I, are repeated for each general service executor that is started. The number of executors started depends on the value you specified on the GSTASK keyword of the OPCOPTS initialization statement. The default is to start all five executors.

Active data-router-task messages

```
EQQZ005I OPC SUBTASK DATA ROUTER TASK IS BEING STARTED
EQQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE
```

When you start a controller and no current plan exists, you will still see a number of EQQZ005I messages each indicating that a subtask is being started. But these subtasks will not start until a current plan is created. You will also see this message:

Current plan message

```
EQQN105W NO VALID CURRENT PLAN EXISTS. CURRENT PLAN VSAM I/O IS NOT
POSSIBLE
```

If you have specified an event-reader function or NCF connections, these tasks will end if no current plan exists. You can verify the remaining tasks when you have created a current plan and connections can be established. “Verifying configuration” on page 174 describes these tasks.

- Check that the log is complete.
“Sample Message Log for a controller” on page 177 shows an example of the MLOG for a controller.

Note: If your log seems to be incomplete, information might be in a buffer. If you are unsure whether the log is complete, issue a dummy modify command like this: F sname,xx. Message EQQZ049E is written to the log when the command is processed. This message will be the last entry in the log.

Checking the server message log

After the controller is started, it starts the servers automatically if you specified the SERVERS keyword on the OPCOPTS statement. Otherwise, you must start them using the z/OS START command. When the server is started, check the message log:

- Ensure that the return code for all the initialization options is 0 (message EQQZ016I)
- Look for these messages when the server is started:

Active server messages

```
EQQZ0051 OPC SUBTASK SERVER IS BEING STARTED
EQQPH001 SERVER TASK HAS STARTED
```

Checking dialog functions

Before invoking the IBM Workload Scheduler for z/OS dialog, ensure that you have set up the ISPF environment as described in “The ISPF environment” on page 188. Then invoke the IBM Workload Scheduler for z/OS dialog, and select an option from 0 to 10 on the main menu. If a new panel appears, you have established communication with the IBM Workload Scheduler for z/OS subsystem. You can further test the dialogs by performing functions, such as creating an application description. For more information on specific dialog functions, see *IBM Workload Scheduler for z/OS: Managing the Workload*.

If you have used RACF to protect controller resources from unauthorized access, verify that the protection mechanism works as expected.

Performing problem determination

If you encounter problems during your verification of the IBM Workload Scheduler for z/OS, correct the errors and verify that the problem has been fixed. For more information on problem determination, see *IBM Workload Scheduler for z/OS: Diagnosis Guide and Reference*.

Dialog problems

Various errors can occur when you are running IBM Workload Scheduler for z/OS dialogs. These errors cause the terminal alarm to sound and a short message to appear in the upper-right corner of your terminal screen. The message text for errors that cause the terminal alarm to sound usually contains the ALARM=YES flag. If this happens when you are trying to verify that IBM Workload Scheduler for z/OS dialogs are correctly installed, press the Help key (usually PF1) in ISPF. ISPF then displays a more complete error message in the long message area on your terminal. The following examples show two dialog error messages. The message number in each example is followed by the long message text and an explanation of the error. The examples highlight two errors. They are related to the dialog interface module, EQQMINOx.

EQQX115

```
EQQX115E TSO Service Facility RC: 20, RSNC: 40
```

The EQQMINOx load module is not installed in a library that can be reached by TSO. EQQMINOx must be present either in the STEPLIB library of the current TSO session or in a library in the current LINKLIB LNKLST nm concatenation. If EQQMINOx has been installed in a LINKLIB library, either an LLA refresh process or an IPL is required to make the module accessible by z/OS users.

EQQX120

The EQQMINOx program can only be called by an APF-authorized task

The EQQMINOx load module must be APF authorized. It must reside in a data set that is defined in SYS1.PARMLIB as being an APF-authorized library. Also, EQQMINOx must be defined to TSO as being an APF-authorized program. Ensure that you have followed the instructions in “Modifying TSO parameters” on page 90.

Authority problems

It is easiest to verify that the controller has been correctly installed without activating authority checking. Then, when authority checking is activated, some TSO users should no longer be able to do what they could do before. An IBM Workload Scheduler for z/OS dialog message should be issued, specifying that they are not authorized to perform a particular dialog function, or that they are not authorized to use any IBM Workload Scheduler for z/OS dialog.

If the controller authority functions have not been correctly installed, this will usually enable TSO users to use dialog functions that they are not authorized to use. The symptom of this problem is the absence of an expected error message. If this happens, follow this procedure which assumes the security monitor being used is RACF.

1. Verify the APPL class. If the user should not be able to use any IBM Workload Scheduler for z/OS dialog, verify that the APPL class is active and that the controller is defined as a resource in the APPL class. Also, verify that the user is not present in the access list to any of these resources and that universal access NONE has been specified. Use the SETR LIST command to display active classes, and use the RLIST command to display access lists in the APPL resource class.
2. Verify that the name of the IBM Workload Scheduler for z/OS RACF resource class has been defined to the IBM Workload Scheduler for z/OS started task in the AUTHDEF statement. You can check this by browsing the controller message log (EQQMLOG).
3. Verify the definition of the IBM Workload Scheduler for z/OS resource class. Check the source of the RACF class descriptor table, and compare this with the definition supplied by the ICHRRRCDE sample in the SEQQSAMP library (see Appendix A, "Sample library (SEQQSAMP)," on page 355).
4. Verify fixed resources. If the user should not be able to use a specific dialog, such as the Calendar dialog, verify that the IBM Workload Scheduler for z/OS RACF resource class is active and that CL is defined as a resource in this class. Also, verify that the user is not present in the access list to the CL resource and that universal access NONE has been specified.
5. Verify subresources. If, for example, the user should be able to update only a subset of all applications in the Application Description dialog, but is instead able to update all applications, verify that the SUBRESOURCES keyword has been correctly specified for the controller in the AUTHDEF statement. Also verify that the controller has been restarted since the AUTHDEF statement was changed, and that IBM Workload Scheduler for z/OS RACF profiles have been refreshed since the IBM Workload Scheduler for z/OS subresource profiles were updated.

Verifying installation of a standby controller

When you have completed the installation tasks for a standby controller, perform initial verification. Because connections cannot be verified in isolation, you can perform further verification of the standby controller when you have installed the controlling system, established connections between IBM Workload Scheduler for z/OS systems, and created a current plan. "Verifying configuration" on page 174 describes these verification tasks.

To initially verify the standby controller, perform these tasks:

1. Ensure that you have completed all the necessary installation tasks.
2. Start the standby controller, and check the message log (EQQMLOG).

Ensuring that all installation tasks are complete

Check the installation tasks to make sure that you have performed the following actions:

- Created a JCL procedure for the standby controller
- Given the security authority for the address space to access the same data sets as the controller
- Specified the initialization statements in the parameter library (EQQPARM)
- Included the standby controller in the same XCF group as the controller
- Defined a VTAM application ID for the standby controller, and activated the VTAM resources, if the standby controller uses NCF connections
- Assigned an IP address to the tracker, if the tracker uses a TCP/IP connection.
- Updated SYS1.PARMLIB and defined VTAM resources, if the IBM Workload Scheduler for z/OS API or the IBM Workload Scheduler for z/OS server is used.

Checking the message log (EQQMLOG)

Start the standby controller.

When the controller has started, you should check the message log.

When you browse the message log:

- Ensure that the return code for all initialization options is 0 (message EQQZ016I).
- Check that this message appears:
Standby controller message
EQQZ128I OPC ACTIVE IN STANDBY MODE

Figure 26 on page 172 shows an example of the MLOG for a standby controller.

```

02/14 13.42.33 EQQZ403I RUNNING TWS for z/OS 9.3.0
02/14 13.42.33 EQQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER STANDBY
02/14 13.42.33 EQQZ015I INIT STATEMENT: OPCOPTS  OPCHOST(STANDBY)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          APPCTASK(YES)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          RECOVERY(YES)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          ERDRTASK(0)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          EWTRTASK(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          GTABLE(JOBCARD)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          NCFTASK(YES)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          NCFAPPL(NCFFN002)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          SERVERS(OSR1,OSR2)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          BUILDSSX(REBUILD)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          VARSUB(SCAN)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          SSCMNAME(EQQSSCMM,TEMPORARY)
02/14 13.42.33 EQQZ015I INIT STATEMENT:
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.33 EQQZ015I INIT STATEMENT: ALERTS  MLOG(DURATION,ERROROPER,OPCERROR,QLIMEXCEED)
02/14 13.42.33 EQQZ015I INIT STATEMENT:
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.33 EQQZ015I INIT STATEMENT: AUTHDEF  CLASS(IBMOPC) SUBRESOURCES(CP.CPGDDEF LT.LTGDEF)
02/14 13.42.33 EQQZ015I INIT STATEMENT:
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.33 EQQZ015I INIT STATEMENT: EXITS   CALL00(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL01(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL02(YES)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL03(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL04(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL05(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL06(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL07(NO)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          CALL09(YES)
02/14 13.42.33 EQQZ015I INIT STATEMENT:
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.33 EQQZ015I INIT STATEMENT: ROUTOPTS DASD(SUBCPU1)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          SNA(NCFFN003)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          USER(OS2LAN1,OS2LAN2)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          XCF(OPC)
02/14 13.42.33 EQQZ015I INIT STATEMENT:
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.33 EQQZ015I INIT STATEMENT: XCFOPTS MEMBER(SMOPC)
02/14 13.42.33 EQQZ015I INIT STATEMENT:          GROUP(PLEXM101)
02/14 13.42.33 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 13.42.34 EQQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER STANDBY  IS: 0000
02/14 13.42.35 EQQZ172I SSX BLOCK OF VERSION 09HWSZ930  SUCCESSFULLY BUILT
02/14 13.42.35 EQQZ073I THE SCHEDULER HAS RECOGNIZED THAT THIS IS A JES2 SYSTEM WITH
02/14 13.42.35 EQQZ073I COMMAND CHARACTER $ AND THAT THE NJE NODE NAME IS TIVZOS
02/14 13.42.35 EQQZ128I SCHEDULER ACTIVE IN STANDBY MODE

```

Figure 26. Sample message log for a standby controller

Verifying installation of the Restart and Cleanup function

To verify that the Restart and Cleanup function was installed and configured correctly, perform these tasks:

- Verify that for each spool a Data Store was installed and started correctly (verify the message log EQQMLOG).
- Verify that the controller was started with the correct parameters (see “SNA only connection” on page 30 for a sample configuration).

Checking the message log (EQQMLOG)

After the controller has been started, ensure that the following messages appear in the message log (this example shows messages for an SNA connection):

```
02/07 12.11.39 EQQZ015I INIT STATEMENT: RCLOPTS CLNJOBPX(EQQCL)
02/07 12.11.39 EQQZ015I INIT STATEMENT:          DSTDEST(TWSFDEST)
02/07 12.11.43 EQQPS01I PRE SUBMITTER TASK INITIALIZATION COMPLETE
02/07 12.11.46 EQQFSF1I DATA FILE EQQSDF01 INITIALIZATION COMPLETED
02/07 12.11.46 EQQFSF1I DATA FILE EQQSDF02 INITIALIZATION COMPLETED
02/07 12.11.46 EQQFSF1I DATA FILE EQQSDF03 INITIALIZATION COMPLETED
02/07 12.11.46 EQQFSI1I SECONDARY KEY FILE INITIALIZATION COMPLETED
02/07 12.11.46 EQQFSD5I SYSOUT DATABASE INITIALIZATION COMPLETE
02/07 12.11.46 EQQFL01I JOBLLOG FETCH TASK INITIALIZATION COMPLETE
02/07 12.11.46 EQQFSD1I SYSOUT DATABASE ERROR HANDLER TASK STARTED
02/07 12.11.46 EQQFV36I SESSION I9PC33A3-I9PC33Z3 ESTABLISHED
```

Note:

1. There should be an EQQFSF1I message for each EQQSDFxx file specified in the startup procedure.
2. There should be an EQQFV36I message for each SNA connection.
3. Verify that the DSTDEST for message EQQZ015I matches the SYSDEST in the Data Store message log.

After the server has been started, ensure that the following messages appear in the message log:

```
02/07 20.16.10 EQQZ015I INIT STATEMENT:          SYSDEST(TWSFDEST)
02/07 20.16.16 EQQFSK1I PRIMARY KEY FILE INITIALIZATION COMPLETED
02/07 20.16.16 EQQFCM2I Data Store COMMAND TASK IS BEING STARTED
02/07 20.16.16 EQQFCC1I Data Store COMMUNICATION TASK INITIALIZATION COMPLETED
02/07 20.16.16 EQQFV01I FN APPLICATION STARTED
02/07 20.16.16 EQQFV24I ACB SUCCESSFULLY OPENED
02/07 20.16.16 EQQFSF1I DATA FILE EQQSDF01 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFV36I SESSION I9PC33Z3-I9PC33A3 ESTABLISHED
02/07 20.16.16 EQQFSF1I DATA FILE EQQSDF02 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSF1I DATA FILE EQQSDF03 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSF1I DATA FILE EQQUDF01 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSF1I DATA FILE EQQUDF02 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSF1I DATA FILE EQQUDF03 INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSI1I SECONDARY KEY FILE INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSD5I SYSOUT DATABASE INITIALIZATION COMPLETE
02/07 20.16.16 EQQFSD1I SYSOUT DATABASE ERROR HANDLER TASK STARTED
02/07 20.16.16 EQQFCU1I CLEAN UP TASK STARTED
02/07 20.16.16 EQQFSW1I Data Store WRITER TASK INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSW1I Data Store WRITER TASK INITIALIZATION COMPLETED
02/07 20.16.16 EQQFSW1I Data Store WRITER TASK INITIALIZATION COMPLETED
02/07 20.16.16 EQQFJK3I Data Store JESQUEUE TASK INITIALIZATION COMPLETED
02/07 20.16.21 EQQFSR1I Data Store READER TASK INITIALIZATION COMPLETED
```

Note:

1. There should be an EQQFSF1I message for each EQQSDFxx file specified in the startup procedure.
2. Verify that the SYSDEST for message EQQZ015I matches the DSTDEST in the controller message log.
3. There should be an EQQFSW1I message for every writer task.
4. There should be an EQQFCC1I message to indicate that the communication completed successfully.

Verifying configuration

When you have installed your IBM Workload Scheduler for z/OS controlling system, or when you install a controlled system, review this section to complete the verification of IBM Workload Scheduler for z/OS. These topics are described:

- Creating entries in the databases
- Running IBM Workload Scheduler for z/OS batch jobs
- Checking the message logs (EQQMLOG)
- Verifying workload submission
- Verifying takeover by a standby controller

Creating entries in the databases

You cannot fully verify IBM Workload Scheduler for z/OS until you have created a current plan. Before you can do this, you must create entries in the databases and then produce a long-term plan. If you are not familiar with IBM Workload Scheduler for z/OS, see *IBM Workload Scheduler for z/OS: Managing the Workload* for information about updating the databases and producing a long-term plan and a current plan.

Sample IBM Workload Scheduler for z/OS databases come with IBM Workload Scheduler for z/OS. They are loaded into the SMP/E target library with ddname SEQQDATA when the tracker software tape is processed. You can load the sample databases by submitting the EQQSAMPI JCL, which is generated by EQQJOBS.

Running batch jobs

When you have created database entries, invoke the LTP dialog and create a long-term plan. Check that the batch job completed successfully, and browse the long-term plan to check that the entries are correct. Next, use the Daily Planning dialog to create a current plan. When this job has ended, browse the current plan to check that the expected application occurrences are present.

You can further test IBM Workload Scheduler for z/OS batch jobs by, for example, printing information about the entries you have created in the databases. Table 20 on page 73 lists the IBM Workload Scheduler for z/OS batch jobs.

Checking the message logs (EQQMLOG)

When you have created a current plan and have started all IBM Workload Scheduler for z/OS address spaces in your configuration, check the message log of the controller and of all trackers.

Controller message log

Look for these messages in the message log of the controller:

Active normal-mode manager messages

```
EQZ005I SUBTASK NORMAL MODE MGR IS BEING STARTED
EQN013I NOW PROCESSING PARAMETER LIBRARY MEMBER EQQCP1DS
```

Note: Active job-tracking log archiver messages. In the preceding message, the active current plan ddname is either EQQCP1DS or EQQCP2DS.

```
EQZ005I SUBTASK JT LOG ARCHIVER IS BEING STARTED
EQN080I THE LOG ARCHIVER TASK HAS STARTED
```

Active job-tracking log archiver messages

Note: The preceding messages, EQQZ085I and EQQG001I, are repeated for each general service executor that is started. The number of executors started depends on the value you specified on the GSTASK keyword of the OPCOPTS initialization statement. The default is to start all five executors.

Active data-router-task messages

```
EQQZ005I SUBTASK DATA ROUTER TASK IS BEING STARTED
EQQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE
```

If you have specified that APPC support should be started, check that these messages appear:

Active APPC-task messages

```
EQQZ005I SUBTASK APPC TASK IS BEING STARTED
EQQ0001I APPC TASK INITIALIZATION IS COMPLETE
```

This message must be issued for the first controller or server start after APPC starts; it is issued by APPC to the system log:

APPC scheduler active - system log messages

```
ATB050I LOGICAL UNIT IS4MEOP4 FOR TRANSACTION SCHEDULER EOP4 HAS BEEN
ADDED TO THE APPC CONFIGURATION.
```

If you have specified an event-reader function, check that these messages appear:

Active event-reader messages

```
EQQZ005I SUBTASK EVENT READER IS BEING STARTED
EQQR025I ERDR 01 STARTED
```

The numeric value on message EQQR025I indicates which event reader is started. The same value cannot be specified on more than one ERSEQNO keyword at the same node. No more than 16 event-reader tasks can be specified at the same node.

If the controller uses XCF connections, the XCF group is activated when the controller is started. Several messages can appear in the message log, indicating that a tracker or standby controller has started and that it has joined the group. If the controller communicates with a tracker using XCF, check for this message to verify the connection:

Active tracker-connection message

```
EQQF007I XCF MEMBER TRACK2 HAS JOINED THE GROUP. THE DESTINATION WILL BE
EQQF007I REPORTED ACTIVE
```

If a standby controller is started, check for this message:

Active standby-controller-connection message

```
EQQF008I XCF MEMBER CTRSTBY1 HAS JOINED THE GROUP AS STANDBY FOR THE
EQQF008I CONTROLLER
```

If the controller uses an NCF connection, check that these messages appear (where NCFCON01 is the VTAM application ID of the controller, and NCFTRK01 is the VTAM application ID of the tracker):

Active NCF-connection messages

```
EQQZ005I SUBTASK VTAM I/O TASK IS BEING STARTED
EQQV001I NCF APPLICATION STARTED
EQQV024I ACB SUCCESSFULLY OPENED
EQQV036I SESSION NCFCON01-NCFTRK01 ESTABLISHED
```

If the controller uses the end-to-end with fault tolerance capabilities feature to schedule on distributed environments, check that these messages appear in the controller EQQMLOG:

Messages for active end-to-end scheduling with fault tolerance capabilities

```
EQQZ005I SUBTASK END TO END ENABLER IS BEING STARTED
EQQZ085I SUBTASK END TO END SENDER IS BEING STARTED
EQQZ085I SUBTASK END TO END RECEIVER IS BEING STARTED
EQQG001I SUBTASK END TO END ENABLER HAS STARTED
EQQG001I SUBTASK END TO END RECEIVER HAS STARTED
EQQG001I SUBTASK END TO END SENDER HAS STARTED
```

When the end-to-end server is started, with the properties file customized to enable all EQQPT messages to be issued to the Server MLOG by default, check that these messages appear in the server EQQMLOG:

Messages in the server for active end-to-end scheduling with fault tolerance capabilities

```
EQQPH33I THE END-TO-END PROCESSES HAVE BEEN STARTED
EQQPT01I Program "/usr/lpp/TWS/TWS930/bin/translator" has been started,
pid is pid number
EQQPT15I The USS bindir "/usr/lpp/TWS/TWS930" maintenance level
is maintenance level
EQQPT01I Program "/usr/lpp/TWS/TWS930/bin/netman" has been started, pid is pid num
```

If a Symphony file has been created and is active, these messages will follow:

```
EQQPT20I Input Translator waiting for Batchman and Mailman are started
EQQPT21I Input Translator finished waiting for Batchman and Mailman
```

Otherwise, if the Symphony file is not present or a new one must be produced, this message will follow:

```
EQQPT22I Input Translator thread stopped until new Symphony will be available
```

The first time that the controller is being started with the fault-tolerant end-to-end scheduling in use or after the event data sets (EQQTWSIN and EQQTWSOU) have been reallocated, the event data sets need to be formatted. The following messages appear in the controller EQQMLOG before the messages about sender and receiver have started:

```
EQQW030I A DISK DATA SET WILL BE FORMATTED, DDNAME = EQQTWSIN
EQQW030I A DISK DATA SET WILL BE FORMATTED, DDNAME = EQQTWSOU
EQQW038I A DISK DATA SET HAS BEEN FORMATTED, DDNAME = EQQTWSIN
EQQW038I A DISK DATA SET HAS BEEN FORMATTED, DDNAME = EQQTWSOU
```

Also, the following messages might appear in the server EQQMLOG:

```
EQQPT56W The /DD:EQQTWSIN queue has not been formatted yet
EQQPT56W The /DD:EQQTWSOU queue has not been formatted yet
```

If the controller uses the Restart and Clean Up functionality check that the following messages appear in the Controller MLOG:

```

EQQZ005I SUBTASK FL TASK IS BEING STARTED
EQQZ005I SUBTASK PRE-SUBMIT IS BEING STARTED
EQQFSD1I SYSOUT DATABASE ERROR HANDLER TASK STARTED
EQQFSK1I PRIMARY KEY FILE INITIALIZATION COMPLETED
EQQFSF1I DATA FILE EQQSDFO1 INITIALIZATION COMPLETED
EQQFSF1I DATA FILE EQQSDFO2 INITIALIZATION COMPLETED
...
EQQFSF1I DATA FILE EQQSDFnn INITIALIZATION COMPLETED
EQQFS11I SECONDARY KEY FILE INITIALIZATION COMPLETED
EQQFSD5I SYSOUT DATABASE INITIALIZATION COMPLETE
EQQPS01I PRE SUBMITTER TASK INITIALIZATION COMPLETE
EQQFL01I JOBLOG FETCH TASK INITIALIZATION COMPLETE

```

If the XCF is used to connect with Data Store Following messages should occur:

```

EQQFCCA1 XCF JOIN STARTED
EQQFCC9I XCF XCFCLC02 HAS JOINED XCF GROUP OPCGRPQ

```

If the SNA is used to connect with Data Store following messages should occur:

```

EQQFV01I FN APPLICATION STARTED
EQQFV24I ACB SUCCESSFULLY OPENED
EQQFV36I SESSION I9PC45RA-I9PC45AA ESTABLISHED

```

Sample Message Log for a controller

Sample Message Log for a controller shows an example of the MLOG for a controller. The controller is connected to three trackers through shared DASD, XCF, and NCF. A standby controller is also started in this configuration.

```

02/15 12.13.19 EQQZ403I RUNNING IWS for z/OS
02/15 12.13.19 EQQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER CONTROLR
02/15 12.13.19 EQQZ015I INIT STATEMENT: OPCOPTS OPCHOST(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: APPCTASK(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: RECOVERY(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: RODMTASK(YES) RODMPARM(RODM)
02/15 12.13.19 EQQZ015I INIT STATEMENT: ERDRTASK(0)
02/15 12.13.19 EQQZ015I INIT STATEMENT: EWTRTASK(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: GTABLE(JOBCARD)
02/15 12.13.19 EQQZ015I INIT STATEMENT: NCFTASK(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: NCFAPPL(NCFFN002)
02/15 12.13.19 EQQZ015I INIT STATEMENT: SERVERS(OSR1,OSR2)
02/15 12.13.19 EQQZ015I INIT STATEMENT: BUILDSSX(REBUILD)
02/15 12.13.19 EQQZ015I INIT STATEMENT: VARSUB(SCAN)
02/15 12.13.19 EQQZ015I INIT STATEMENT: SSCMNAME(EQQSSCMM,TEMPORARY)
02/15 12.13.19 EQQZ015I INIT STATEMENT:
02/15 12.13.19 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQQZ015I INIT STATEMENT: ALERTS MLOG(DURATION,ERROROPER,OPCERROR,QLIMEXCEED,RESCONT)
02/15 12.13.19 EQQZ015I INIT STATEMENT:
02/15 12.13.19 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQQZ015I INIT STATEMENT: AUTHDEF CLASS(IBMOPC) SUBRESOURCES(CP.CPGDDEF LT.LTGDEF)
02/15 12.13.19 EQQZ015I INIT STATEMENT:
02/15 12.13.19 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQQZ015I INIT STATEMENT: EXITS CALL00(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL01(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL02(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL03(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL04(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL05(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL06(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL07(NO)
02/15 12.13.19 EQQZ015I INIT STATEMENT: CALL09(YES)
02/15 12.13.19 EQQZ015I INIT STATEMENT:
02/15 12.13.19 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQQZ015I INIT STATEMENT: ROUTOPTS DASD(SUBCPU1)
02/15 12.13.19 EQQZ015I INIT STATEMENT: SNA(NCFFN003)
02/15 12.13.19 EQQZ015I INIT STATEMENT: USER(OS2LAN1,OS2LAN2)
02/15 12.13.19 EQQZ015I INIT STATEMENT: XCF(SMOPC)
02/15 12.13.19 EQQZ015I INIT STATEMENT: APPC(ROMEAS1:ITIBM200.S44D1288)
02/15 12.13.19 EQQZ015I INIT STATEMENT: TCP(ROM2:9.52.52.11)
02/15 12.13.19 EQQZ015I INIT STATEMENT: TCP1PID(TCPIPROC)
02/15 12.13.19 EQQZ015I INIT STATEMENT:

```

```

02/15 12.13.19 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQZ015I INIT STATEMENT: XCOPTS MEMBER(OPC)
02/15 12.13.19 EQZ015I INIT STATEMENT: GROUP(PLEXM101)
02/15 12.13.19 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.19 EQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER CONTROLR IS: 0000
02/15 12.13.20 EQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER RODM
02/15 12.13.20 EQZ015I INIT STATEMENT: RODMOPTS RODMSYSTEM(EKGXROM)
02/15 12.13.20 EQZ015I INIT STATEMENT: OPCRESOURCE(TAPE)
02/15 12.13.20 EQZ015I INIT STATEMENT: OPCFIELD(QUANTITY)
02/15 12.13.20 EQZ015I INIT STATEMENT: RODMCLASS(TAPE)
02/15 12.13.20 EQZ015I INIT STATEMENT: RODMFIELD(NUMBEROF)
02/15 12.13.20 EQZ015I INIT STATEMENT: RODMLOST(LAST)
02/15 12.13.20 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.20 EQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER ROOM
02/15 12.13.21 EQZ172I SSX BLOCK OF VERSION 09HWSZ930 SUCCESSFULLY BUILT
02/15 12.13.21 EQZ073I THE SCHEDULER HAS RECOGNIZED THAT THIS IS A JES2 SYSTEM WITH
02/15 12.13.21 EQZ073I COMMAND CHARACTER $ AND THAT THE NJE NODE NAME IS TIVZOS
02/15 12.13.22 EQZ005I SUBTASK VTAM I/O TASK IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK NORMAL MODE MGR IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK TCP/IP TASK IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK APPC TRACKER IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK JOB SUBMIT TASK IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK DATA ROUTER TASK IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK RODM TASK IS BEING STARTED
02/15 12.13.22 EQZ005I SUBTASK APPC TASK IS BEING STARTED
02/15 12.13.23 EQV001I NCF APPLICATION STARTED
02/15 12.13.23 EQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER CONTROLR
02/15 12.13.23 EQZ015I INIT STATEMENT: AUDIT ACCESS(UPDATE) AMOUNT(KEY) FILE(ALL)
02/15 12.13.23 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.23 EQZ015I INIT STATEMENT: AUDIT ACCESS(UPDATE) AMOUNT(DATA) FILE(JS)
02/15 12.13.23 EQZ015I INIT STATEMENT:
02/15 12.13.23 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.23 EQZ015I INIT STATEMENT: JTOPTS BACKUP(NO)
02/15 12.13.23 EQZ015I INIT STATEMENT: STATMSG(CPLOCK EVENTS GENSERV)
02/15 12.13.23 EQZ015I INIT STATEMENT: ETT(YES)
02/15 12.13.23 EQZ015I INIT STATEMENT: HIGHRC(0)
02/15 12.13.23 EQZ015I INIT STATEMENT: JOBCHECK(SAME)
02/15 12.13.23 EQZ015I INIT STATEMENT: JTLOGS(5)
02/15 12.13.23 EQZ015I INIT STATEMENT: OPINFOSCOPE(ALL)
02/15 12.13.23 EQZ015I INIT STATEMENT: JOBSUBMIT(YES)
02/15 12.13.23 EQZ015I INIT STATEMENT: MAXJSFILE(NO)
02/15 12.13.23 EQZ015I INIT STATEMENT: NEWOILIMIT(30)
02/15 12.13.23 EQZ015I INIT STATEMENT: OFFDELAY(3)
02/15 12.13.23 EQZ015I INIT STATEMENT: PLANSTART(7)
02/15 12.13.23 EQZ015I INIT STATEMENT: PRTCOMPLETE(YES)
02/15 12.13.23 EQZ015I INIT STATEMENT: DUAL(NO)
02/15 12.13.23 EQZ015I INIT STATEMENT: SHUTDOWNPOLICY(75)
02/15 12.13.23 EQZ015I INIT STATEMENT: SUBFAILURE(E)
02/15 12.13.23 EQZ015I INIT STATEMENT: SUPPRESSACTION(C)
02/15 12.13.23 EQZ015I INIT STATEMENT: SUPPRESSPOLICY(75)
02/15 12.13.23 EQZ015I INIT STATEMENT: TRACK(JOBOPT)
02/15 12.13.23 EQZ015I INIT STATEMENT: WSFAILURE(ERROR,REROUTE,IMMED)
02/15 12.13.23 EQZ015I INIT STATEMENT: WSOFFLINE(ERROR,REROUTE,IMMED)
02/15 12.13.23 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.23 EQZ015I INIT STATEMENT: NOERROR LIST(MYJOB.*.STEP4.0002
02/15 12.13.23 EQZ015I INIT STATEMENT: HISJOB.*.STEP6.0004
02/15 12.13.23 EQZ015I INIT STATEMENT: HERJOB.*.STEP9.0016)
02/15 12.13.23 EQZ015I INIT STATEMENT:
02/15 12.13.23 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.23 EQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER CONTROLR IS: 0000
02/15 12.13.23 EQSU12I MAX NUMBER OF WORKSTATIONS CHECKPOINTED BY THIS SUBMIT TASK: 0046
02/15 12.13.23 EQQ001I APPC TASK INITIALIZATION IS COMPLETE
02/15 12.13.23 EQF007I XCF MEMBER SMOPC HAS JOINED THE GROUP. THE DESTINATION WILL BE
02/15 12.13.23 EQF007I REPORTED ACTIVE
02/15 12.13.23 EQF008I XCF MEMBER SBOPC HAS JOINED THE GROUP AS STANDBY FOR THE
02/15 12.13.23 EQF008I CONTROLLER
02/15 12.13.23 EQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE
02/15 12.13.23 EQTA01I THE TCP/IP COMMUNICATION TASK HAS STARTED
02/15 12.13.24 EQSU01I THE SUBMIT TASK HAS STARTED
02/15 12.13.24 EQAT01I THE APPC TRACKER TASK HAS STARTED
02/15 12.13.26 EQQ502I SPECIAL RESOURCE DATASPACE HAS BEEN CREATED
02/15 12.13.26 EQQ502I 0000020 PAGES ARE USED FOR 00000100 SPECIAL RESOURCE RECORDS
02/15 12.13.45 EQN018I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY ALLOCATED FOR VSAM FILE EQQCP1DS
02/15 12.13.45 EQN018I NUMBER OF INDEX BUFFERS ARE 000006 WITH SIZE 016384
02/15 12.13.45 EQN018I NUMBER OF DATA BUFFERS ARE 000010 WITH SIZE 032768
02/15 12.13.45 EQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE EQQJT01
02/15 12.13.45 EQN013I JOB TRACKING IS NOW ACTIVE AND CURRENT PLAN DD-NAME IS EQQCP1DS
02/15 12.13.45 EQZ005I SUBTASK EVENT MANAGER IS BEING STARTED
02/15 12.13.45 EQZ005I SUBTASK GENERAL SERVICE IS BEING STARTED
02/15 12.13.45 EQZ005I SUBTASK AUTO RECOVERY IS BEING STARTED
02/15 12.13.45 EQZ005I SUBTASK JT LOG ARCHIVER IS BEING STARTED
02/15 12.13.45 EQZ005I SUBTASK EXTERNAL ROUTER IS BEING STARTED
02/15 12.13.45 EQZ005I SUBTASK WS ANALYZER IS BEING STARTED
02/15 12.13.45 EQN080I THE LOG ARCHIVER TASK HAS STARTED
02/15 12.13.46 EQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER STDAR
02/15 12.13.46 EQZ015I INIT STATEMENT: AROPTS PREDWS(CPU*)
02/15 12.13.46 EQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/15 12.13.46 EQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER STDAR IS: 0000

```

```

02/15 12.13.46 EQW505I THE WORK STATION ANALYZER TASK HAS STARTED
02/15 12.13.46 EQZ085I SUBTASK GS EXECUTOR 01 IS BEING STARTED
02/15 12.13.46 EQC001I THE AUTOMATIC RECOVERY SUBTASK HAS STARTED

02/15 12.13.46 EQZ085I SUBTASK GS EXECUTOR 02 IS BEING STARTED
02/15 12.13.46 EQEX01I THE EXTERNAL ROUTER TASK HAS STARTED
02/15 12.13.46 EQZ085I SUBTASK GS EXECUTOR 03 IS BEING STARTED
02/15 12.13.46 EQZ085I SUBTASK GS EXECUTOR 04 IS BEING STARTED
02/15 12.13.46 EQZ085I SUBTASK GS EXECUTOR 05 IS BEING STARTED
02/15 12.13.46 EQG001I SUBTASK GENERAL SERVICE HAS STARTED
02/15 12.13.46 EQE025I THE EVENT MANAGER HAS STARTED
02/15 12.13.46 EQE017I THE ETT FUNCTION IS ACTIVATED
02/15 12.13.46 EQG001I SUBTASK GS EXECUTOR 01 HAS STARTED
02/15 12.13.46 EQG001I SUBTASK GS EXECUTOR 02 HAS STARTED
02/15 12.13.46 EQG001I SUBTASK GS EXECUTOR 03 HAS STARTED
02/15 12.13.46 EQG001I SUBTASK GS EXECUTOR 04 HAS STARTED
02/15 12.13.46 EQG001I SUBTASK GS EXECUTOR 05 HAS STARTED
02/15 12.17.29 EQV024I ACB SUCCESSFULLY OPENED
02/15 12.17.59 EQV036I SESSION NCFN002-NCFN003 ESTABLISHED
02/15 12.20.35 EQN051I A CURRENT PLAN BACKUP PROCESS HAS STARTED. TRIGGER WAS: BACKUP CMD
02/15 12.20.35 EQN012I JOB TRACKING EVENTS ARE NOW BEING LOGGED ON FILE EQJT02
02/15 12.20.35 EQQ0507I A SPECIAL RESOURCE DATASPACE BACKUP PROCESS HAS STARTED
02/15 12.20.35 EQQ0508I A SPECIAL RESOURCE DATASPACE BACKUP PROCESS HAS ENDED
02/15 12.20.35 EQQ0508I 00000003 RECORDS WERE WRITTEN TO CX
02/15 12.20.38 EQN056I A CURRENT PLAN COPY PROCESS HAS STARTED
02/15 12.20.40 EQN057I A CURRENT PLAN DATA SET WAS SUCCESSFULLY COPIED: FROMDD=EQQCP1DS, TODD=EQQCP2DS
02/15 12.20.40 EQN023I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY DELETED FOR VSAM FILE EQQCP2DS
02/15 12.20.41 EQN018I VSAM LSR BUFFERS HAVE BEEN SUCCESSFULLY ALLOCATED FOR VSAM FILE EQQCP2DS
02/15 12.20.41 EQN018I NUMBER OF INDEX BUFFERS ARE 000006 WITH SIZE 016384
02/15 12.20.41 EQN018I NUMBER OF DATA BUFFERS ARE 000010 WITH SIZE 032768
02/15 12.20.41 EQN090I THE JOB TRACKING LOG DATA SET DEFINED BY DDNAME EQJT01 HAS BEEN
02/15 12.20.41 EQN090I COPIED TO THE JOB TRACKING LOG ARCHIVE DATA SET
02/15 12.20.57 EQZ000I A STOP COMMAND HAS BEEN RECEIVED
02/15 12.20.57 EQN000I THE NORMAL MODE MANAGER TASK HAS BEEN REQUESTED TO TERMINATE
02/15 12.20.57 EQSU02I THE SUBMIT TASK HAS ENDED
02/15 12.20.57 EQEX02I THE EXTERNAL ROUTER TASK HAS ENDED
02/15 12.20.57 EQZ034I SUBTASK DATA ROUTER TASK HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 453 SECONDS AND USED 0.1 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK APPC TASK HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 453 SECONDS AND USED 0.0 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK JOB SUBMIT TASK HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 453 SECONDS AND USED 0.1 CPU SECONDS
02/15 12.20.57 EQE000I TOTAL NUMBER OF EVENTS PROCESSED BY THE EVENT MANAGER TASK IS: 34
02/15 12.20.57 EQE000I NUMBER OF EVENTS SINCE THE PREVIOUS MESSAGE IS: 34
02/15 12.20.57 EQE000I EVENT MANAGER QUEUE LENGTH STATISTICS FOLLOW:
02/15 12.20.57 EQE000I TOTAL Q1 Q2 Q5 Q10 Q20 Q50 Q100 >100
02/15 12.20.57 EQE000I 31 28 2 1 0 0 0 0 0
02/15 12.20.57 EQE006I EVENT MANAGER EVENT TYPE STATISTICS FOLLOW:
02/15 12.20.57 EQE006I TYPE NTOT NNEW TTOT TNEW TAVG NAVG NSUS
02/15 12.20.57 EQE007I ALL 34 34 0.2 0.2 0.00 0.00 0
02/15 12.20.57 EQE007I 1 3 3 0.0 0.0 0.03 0.03 0
02/15 12.20.57 EQE007I 2 3 3 0.0 0.0 0.02 0.02 0
02/15 12.20.57 EQE007I 3S 0 0 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I 3J 4 4 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I 3P 4 4 0.1 0.1 0.02 0.02 0
02/15 12.20.57 EQE007I 4 0 0 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I 5 12 12 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I USER 0 0 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I CATM 0 0 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE007I OTHR 8 8 0.0 0.0 0.00 0.00 0
02/15 12.20.57 EQE004I CP ENQ LOCK STATISTICS SINCE PREVIOUS MESSAGE FOLLOW:
02/15 12.20.57 EQE004I NAME NEXCL NSHRD THELD TWAIT AHELD AWAIT
02/15 12.20.57 EQE005I NORMAL MODE MGR 18 0 7.0 0.0 0.39 0.00
02/15 12.20.57 EQE005I WS ANALYZER 7 0 0.7 0.1 0.10 0.02
02/15 12.20.57 EQE005I EVENT MANAGER 31 0 0.2 5.6 0.00 0.18

02/15 12.20.57 EQV037I SESSION NCFN002-NCFN003 ENDED
02/15 12.20.57 EQC003I THE AUTOMATIC RECOVERY SUBTASK HAS ENDED NORMALLY
02/15 12.20.57 EQTA02I THE TCP/IP COMMUNICATION TASK HAS ENDED
02/15 12.20.57 EQAT01I THE APPC TRACKER TASK HAS ENDED
02/15 12.20.57 EQG010I GENERAL SERVICE REQUEST STATISTICS FOLLOW:
02/15 12.20.57 EQG010I TYPE TOTAL NEWRQS TOTTIME NEWTIME TOTAVG NEWAVG
02/15 12.20.57 EQG011I ALL 0 0 0.0 0.0 0.00 0.00
02/15 12.20.57 EQV020I ACB SUCCESSFULLY CLOSED
02/15 12.20.57 EQZ034I SUBTASK EXTERNAL ROUTER HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 431 SECONDS AND USED 0.0 CPU SECONDS
02/15 12.20.57 EQE023I THE EVENT MANAGER ENDED NORMALLY
02/15 12.20.57 EQW503I THE WORK STATION ANALYZER ENDED NORMALLY
02/15 12.20.57 EQG003I SUBTASK GS EXECUTOR 01 HAS ENDED
02/15 12.20.57 EQG003I SUBTASK GS EXECUTOR 02 HAS ENDED
02/15 12.20.57 EQV006I NCF APPLICATION ENDED
02/15 12.20.57 EQZ034I SUBTASK AUTO RECOVERY HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 431 SECONDS AND USED 0.0 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK TCP/IP TASK HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 431 SECONDS AND USED 0.0 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK APPC TRACKER HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE 431 SECONDS AND USED 0.0 CPU SECONDS
02/15 12.20.57 EQG003I SUBTASK GS EXECUTOR 03 HAS ENDED
02/15 12.20.57 EQG003I SUBTASK GS EXECUTOR 04 HAS ENDED
02/15 12.20.57 EQG003I SUBTASK GS EXECUTOR 05 HAS ENDED
02/15 12.20.57 EQE018I THE ETT FUNCTION IS DEACTIVATED
02/15 12.20.57 EQZ034I SUBTASK VTAM I/O TASK HAS ENDED.

```

```

02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE      454 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK WS ANALYZER HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.4 CPU SECONDS
02/15 12.20.57 EQZ034I SUBTASK EVENT MANAGER HAS ENDED.
02/15 12.20.57 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GS EXECUTOR 01 HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GS EXECUTOR 02 HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GS EXECUTOR 03 HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GS EXECUTOR 04 HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GS EXECUTOR 05 HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      431 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.03 EQZ034I SUBTASK GENERAL SERVICE HAS ENDED
02/15 12.21.03 EQZ034I SUBTASK GENERAL SERVICE HAS ENDED.
02/15 12.21.03 EQZ034I SUBTASK WAS ACTIVE      437 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.05 EQN081I THE LOG ARCHIVER TASK HAS ENDED
02/15 12.21.05 EQZ034I SUBTASK JT LOG ARCHIVER HAS ENDED.
02/15 12.21.05 EQZ034I SUBTASK WAS ACTIVE      439 SECONDS AND USED    0.0 CPU SECONDS
02/15 12.21.05 EQN107I THE NORMAL MODE MANAGER TASK HAS ENDED
02/15 12.21.05 EQZ034I SUBTASK NORMAL MODE MGR HAS ENDED.
02/15 12.21.05 EQZ034I SUBTASK WAS ACTIVE      461 SECONDS AND USED    7.0 CPU SECONDS
02/15 12.21.06 EQZ173I SSX BLOCK OF VERSION 09HWSZ930 SUCCESSFULLY RESTORED
02/15 12.21.08 EQZ006I NO ACTIVE PRODUCT SUBTASKS. THE SCHEDULER IS ENDING

```

Tracker message log

Look for the following messages in the log of each tracker:

Active data-router-task messages

```

EQZ005I SUBTASK DATA ROUTER TASK IS BEING STARTED
EQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE

```

Active submit-task messages

```

EQZ005I SUBTASK JOB SUBMIT TASK IS BEING STARTED
EQSU01I THE SUBMIT TASK HAS STARTED

```

Also, verify that the tracker has started an event writer. You should see these messages:

Active event-writer messages

```

EQZ005I SUBTASK EVENT WRITER IS BEING STARTED
EQW065I EVENT WRITER STARTED

```

If the tracker forwards events to the controller, ensure that an event-reader function is specified. This can be a separate event-reader task or an event writer that has been started with the EWSEQNO keyword. In some configurations, both functions can be started in a tracker.

- Although no messages are issued if you use EWSEQNO to start the reader function, check that EWSEQNO appears on the EWTROPTS statement and that the return code for this statement is 0.
- If you have specified a separate event-reader function, check that these messages appear:

Active event-reader messages

```

EQZ005I SUBTASK EVENT READER IS BEING STARTED
EQR025I ERDR 01 STARTED

```

The numeric value on message EQR025I indicates which event reader is being started. The same value cannot be specified on EWSEQNO and ERSEQNO keywords at the same node, and no more than 16 reader tasks can be specified at this node.

Note: If the tracker uses an XCF connection, no messages appear in the tracker message log unless an error has occurred. To verify XCF connection messages, check the message log of the controller.

If the tracker uses an NCF connection, check that these messages appear (where NCFTRK01 and NCFCON01 represent the VTAM application IDs of the tracker and controller):

Active NCF-connection messages

```

EQQZ005I SUBTASK VTAM I/O TASK IS BEING STARTED
EQQV001I NCF APPLICATION STARTED
EQQV024I ACB SUCCESSFULLY OPENED
EQQV036I SESSION NCFTRK01-NCFCON01 ESTABLISHED
EQQV040I CURRENTLY RUNNING WITH 'NCFCON01' AS CONTROLLER

```

Sample message log for a tracker

Sample message log for a tracker shows an example of the MLOG for a tracker. The tracker is connected to the controller via a VTAM link. The VTAM application IDs are NCFTRK01 for the tracker and NCFCON01 for the controller. The controller is active.

```

02/14 12.14.08 EQQZ403I RUNNING IWS for z/OS 9.3.0
02/14 12.14.08 EQQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER TRKV
02/14 12.14.08 EQQZ015I INIT STATEMENT: OPCOPTS OPCHOST(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: NCFTASK(YES)
02/14 12.14.08 EQQZ015I INIT STATEMENT: NCFAPPL(NCFFN003)
02/14 12.14.08 EQQZ015I INIT STATEMENT: JCCTASK(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: EWTRTASK(YES)
02/14 12.14.08 EQQZ015I INIT STATEMENT: EWTRPARM(WTRPARMS)
02/14 12.14.08 EQQZ015I INIT STATEMENT: ERDRTASK(0)

02/14 12.14.08 EQQZ015I INIT STATEMENT: SSCMNAME(EQSSCMM,TEMPORARY)
02/14 12.14.08 EQQZ015I INIT STATEMENT: BUILDSSX(REBUILD)
02/14 12.14.08 EQQZ015I INIT STATEMENT:
02/14 12.14.08 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.08 EQQZ015I INIT STATEMENT: JOBOPTS CHSMWAIT(1)
02/14 12.14.08 EQQZ015I INIT STATEMENT: JOBGLORETRIEVAL(DELAYED)
02/14 12.14.08 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.08 EQQZ015I INIT STATEMENT: AUTHDEF CLASS(IBMOPC) SUBRESOURCES(RL.WSNAME SR.SRNAME)
02/14 12.14.08 EQQZ015I INIT STATEMENT:
02/14 12.14.08 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.08 EQQZ015I INIT STATEMENT: EXITS CALL00(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL01(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL02(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL03(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL04(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL05(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL06(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL07(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT: CALL09(NO)
02/14 12.14.08 EQQZ015I INIT STATEMENT:
02/14 12.14.08 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.08 EQQZ015I INIT STATEMENT: TRROPTS HOSTCON(SNA)
02/14 12.14.08 EQQZ015I INIT STATEMENT: SNAHOST(NCFFN002)
02/14 12.14.08 EQQZ015I INIT STATEMENT:
02/14 12.14.08 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.08 EQQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER TRKV IS: 0000
02/14 12.14.08 EQQZ172I SSX BLOCK OF VERSION 09HWSZ930 SUCCESSFULLY BUILT
02/14 12.14.08 EQQZ073I THE SCHEDULER HAS RECOGNIZED THAT THIS IS A JES2 SYSTEM WITH
02/14 12.14.08 EQQZ073I COMMAND CHARACTER $ AND THAT THE NJE NODE NAME IS TIVZOS
02/14 12.14.08 EQQZ005I SUBTASK EVENT WRITER IS BEING STARTED
02/14 12.14.08 EQQZ005I SUBTASK VTAM I/O TASK IS BEING STARTED
02/14 12.14.08 EQQZ005I SUBTASK JOB SUBMIT TASK IS BEING STARTED
02/14 12.14.08 EQQZ005I SUBTASK DATA ROUTER TASK IS BEING STARTED
02/14 12.14.09 EQQZ013I NOW PROCESSING PARAMETER LIBRARY MEMBER WTRPARMS
02/14 12.14.09 EQQZ015I INIT STATEMENT: EWTRROPTS HOLDJOB(USER)
02/14 12.14.09 EQQZ015I INIT STATEMENT: EWWAIT(7)
02/14 12.14.09 EQQZ015I INIT STATEMENT: SUREL(NO)
02/14 12.14.09 EQQZ015I INIT STATEMENT: EWSQNO(01)
02/14 12.14.09 EQQZ015I INIT STATEMENT: STEPEVENTS(NZERO)
02/14 12.14.09 EQQZ015I INIT STATEMENT: RETCODE(HIGHEST)

02/14 12.14.09 EQQZ016I RETURN CODE FOR THIS STATEMENT IS: 0000
02/14 12.14.09 EQQZ014I MAXIMUM RETURN CODE FOR PARAMETER MEMBER WTRPARMS IS: 0000
02/14 12.14.09 EQQZ064I THE SCHEDULER WILL USE THE NJE NODE NAME TIVZOS FOR JOBS HELD ON THIS NODE
02/14 12.14.09 EQQV001I NCF APPLICATION STARTED
02/14 12.14.09 EQQZ065I THE SCHEDULER RELEASES HELD JOBS USING THE JES2 COMMAND CHARACTER $
02/14 12.14.09 EQQW026I THE JOB COMPLETION CHECKER STARTED
02/14 12.14.09 EQQF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE
02/14 12.14.09 EQQSU12I MAX NUMBER OF WORKSTATIONS CHECKPOINTED BY THIS SUBMIT TASK: 0046
02/14 12.14.09 EQQSU01I THE SUBMIT TASK HAS STARTED
02/14 12.14.14 EQQW065I EVENT WRITER STARTED
02/14 12.15.59 EQQV033E ACB OPEN FAILED FOR THE LAST 2 MINUTES - NCF APPLICATION NOT ACTIVE
02/14 12.17.49 EQQV024I ACB SUCCESSFULLY OPENED
02/14 12.17.59 EQQV036I SESSION NCFFN003-NCFFN002 ESTABLISHED
02/14 12.18.09 EQQV040I CURRENTLY RUNNING WITH 'NCFFN002' AS CONTROLLER

```

```

02/14 12.22.23 EQZ000I A STOP COMMAND HAS BEEN RECEIVED
02/14 12.22.23 EQW011I THE EVENT WRITER ENDED NORMALLY
02/14 12.22.23 EQV037I SESSION NCFN003-NCFN002 ENDED
02/14 12.22.23 EQSU02I THE SUBMIT TASK HAS ENDED
02/14 12.22.23 EQZ034I SUBTASK DATA ROUTER TASK HAS ENDED.
02/14 12.22.23 EQZ034I SUBTASK WAS ACTIVE      494 SECONDS AND USED      0.0 CPU SECONDS
02/14 12.22.23 EQZ034I SUBTASK WAS ACTIVE      494 SECONDS AND USED      0.0 CPU SECONDS
02/14 12.22.23 EQZ034I SUBTASK EVENT WRITER HAS ENDED.
02/14 12.22.23 EQZ034I SUBTASK WAS ACTIVE      494 SECONDS AND USED      0.1 CPU SECONDS
02/14 12.22.23 EQZ034I SUBTASK JOB SUBMIT TASK HAS ENDED.
02/14 12.22.23 EQZ034I SUBTASK WAS ACTIVE      494 SECONDS AND USED      0.2 CPU SECONDS
02/14 12.22.27 EQV020I ACB SUCCESSFULLY CLOSED
02/14 12.22.27 EQV006I NCF APPLICATION ENDED
02/14 12.22.27 EQZ034I SUBTASK VTAM I/O TASK HAS ENDED.
02/14 12.22.27 EQZ034I SUBTASK WAS ACTIVE      498 SECONDS AND USED      0.0 CPU SECONDS
02/14 12.22.28 EQZ173I SSX BLOCK OF VERSION 09HWSZ930 SUCCESSFULLY RESTORED
02/14 12.22.30 EQZ006I NO ACTIVE PRODUCT SUBTASKS. THE SCHEDULER IS ENDING

```

Verifying workload submission

Now verify that IBM Workload Scheduler for z/OS can submit work and that the work is sent to the correct destination.

Controlling system

You can use this procedure for the controlling system:

1. Create a workstation description, leave the destination field blank. This means that operations will be submitted to the system where the controller is started.
2. Create an application description. Define at least one operation on the workstation you have created. Submit a daily planning extend or replan batch job to include the new workstation in the current plan.
Add an occurrence of this application to the current plan.
3. Verify that the operations run successfully on this system and that they are reported as complete in the current plan.
4. Check that submit events are created in the event data set for the controlling system (see “Verifying job submission” on page 183).

Controlled systems

If you have controlled IBM Workload Scheduler for z/OS systems in your configuration, you can use this procedure to check that work is sent to these destinations, that it is submitted, and that events are returned to the controller:

1. Create a workstation description for each destination. In the destination field, remember to specify the submit/release data set name, the XCF member name of the tracker, the tracker VTAM application ID, or the tracker TCP/IP destination name, depending on the connection. Ensure that the trackers are active.
2. Add an application occurrence to the current plan with operations defined on each of the workstations.
3. Verify that the operations run successfully on the correct system and that they are reported as complete in the current plan.
4. Check that submit events are created in the event data set for each controlled system (see “Verifying job submission” on page 183).

If your configuration includes a MAS complex, you can specify only one workstation description to represent all systems in the complex. If this is the case, ensure that a job is run on each system in the complex. Verify that events are received at the controller by checking that the operations are reported as complete in the current plan.

Note:

1. If you create a workstation description after the current plan has been created, you must either replan or extend the current plan to use this workstation.

2. You must also specify workstation destinations in the ROUTOPTS initialization statement. You must restart the controller if you update ROUTOPTS.

Verifying job submission

When IBM Workload Scheduler for z/OS submits work, a submit event is written to the event data set. A submit event is prefixed with an I, and can be one of these:

- IJ1 Job JCL. A job has been submitted.
- IJ2 Started-task JCL. A started task has been started.
- IWTO An operation has been initiated for a general workstation with the WTO option. The submit task causes message EQQW775I to be issued.

Check each system on which IBM Workload Scheduler for z/OS is installed to ensure that the destination can be reached by the controller and that the relevant submit events are being created in the event data set. That is, if IBM Workload Scheduler for z/OS will submit jobs, start started tasks, and initiate WTO operations, verify that all the event types are created in the event data set. You need not test all these functions if IBM Workload Scheduler for z/OS will not be used for a particular operation.

To perform this test for all type I events, start an operation on each of three workstations, all of which specify the destination of the system you are testing. The workstations must be a computer automatic workstation for IJ1 events, a computer automatic workstation with the started-task option for IJ2 events, and a general WTO workstation for IWTO events. Follow this procedure to verify workload submission:

1. Ensure that you have the correct workstations specified in the workstation description database.
2. Create a test application with an operation for each workstation you want to test, and add it to the current plan.
3. When the application has completed, browse the event data set, and verify that the required type I events have been created.
4. If the operations are not started, check that the workstation status is ACTIVE and that the workstation is not WAITING FOR CONNECTION, which means that the controller is waiting for the corresponding tracker to communicate.

Verifying takeover by a standby controller

To verify that a standby controller can take over the functions of the controller, perform these actions:

1. Stop the controller.
2. Issue the following command:
MODIFY sname,TAKEOVER
For a description about how to set up automatic takeover as a runtime option, see the XCFOPTS statement in *Customization and Tuning*.
3. Check that this message appears in the message log:
Standby controller message
EQZ129I TAKEOVER IN PROGRESS

When the standby controller has taken over the functions of the controller, more messages will appear in the message log. These are the same messages that appear when a controller is started. Verify that the takeover was successful by following the procedures used in the verification of the controller.

Chapter 6. Migrating

This chapter provides information to help you plan your migration from IBM Workload Scheduler for z/OS versions 8.5.1, 8.6, 9.1, and 9.2 to version 9.3.

Note: To migrate from version 9.2 to version 9.3, you must have installed APARs PI28025 and PI28027.

Planning for migration

Before attempting to perform a migration, develop a migration plan to ensure a smooth and orderly transition. A well thought-out and documented migration plan can help minimize any interruption of service. Your migration plan should address topics such as:

- Identifying which required and optional products are needed
- Evaluating new, changed, and deleted functions
- Defining which IBM Workload Scheduler for z/OS functions you want to add, delete, or modify
- Defining necessary changes to:
 - The configuration
 - The initialization statements
 - Installation modifications
 - Operational procedures
 - Other related products
- Determining any restrictions during the conversion period
- Estimating the amount of time the conversion will take
- Defining education requirements for operators and users
- Preparing your staff and users for migration.

The content and extent of a migration plan can vary significantly from installation to installation. For example, installations that have many installation-specific modifications might require extensive planning due to the added complexity.

You should also consider the following areas when defining your migration plan:

- Installation
- Initialization
- Customization
- Operation

Migration considerations

IBM attempts to make the installation of new releases as easy as possible. Initially, you should install IBM Workload Scheduler for z/OS without taking any customization actions in order to achieve a stable environment. Refer to the *Program Directory* for specific instructions about using System Modification Program/Extended (SMP/E) to install IBM Workload Scheduler for z/OS.

You can migrate from or fall back to previous releases **without** IPLing z/OS.

The following are some migration considerations and, in some cases, necessary prerequisite steps to perform, before migrating to the current release, to ensure a proper fallback migration can be performed, if necessary at any time.

- If you are performing a fallback because of problems experienced on IBM Workload Scheduler for z/OS, be sure to keep the IBM Workload Scheduler for z/OS data sets for diagnostic purposes.
- If you migrate to and fallback from IBM Workload Scheduler for z/OS to test the environment before your *official* migration, ensure that you reallocate all IBM Workload Scheduler for z/OS data sets before the next migration exercise.
- Before you migrate to IBM Workload Scheduler for z/OS Version 9.1 from any earlier version, ensure that you have applied at least the fix for PM86419 on the old version. This is required to avoid problems if a fallback migration is necessary.
- It might occur that in the Application Description database, an application shows a timestamp for its Last Update that corresponds to a date when no changes were made. This can occur when you migrate to a recent version of IBM Workload Scheduler for z/OS, because for the new fields introduced, the database conversion program adds them as blank fields; when you access a panel, *every displayed field* is verified and if a field is blank or null, the verification program initializes the field to its default. This is considered a change because different from the original record content. For this reason, the record changed message is issued, and the Last Update date is modified.

Customization considerations

IBM Workload Scheduler for z/OS is designed as a general-purpose workload automation subsystem. As such, it might not meet all the requirements for your specific installation. IBM allows installations to implement installation exits and provides many callable services that can be used to supplement IBM Workload Scheduler for z/OS processing.

Carefully examine any customization that your site has installed. Determine whether the function is now provided in the product or if you need to modify the logic based on changes made to IBM Workload Scheduler for z/OS.

When new functions are added to IBM Workload Scheduler for z/OS, installation exits and macros used within installation exits can change. New installation exits or macros might also be introduced in an IBM Workload Scheduler for z/OS release. If a release provides a new installation exit, determine if your installation needs to implement the exit. A release can change an existing exit by modifying:

- What the installation exit expects on entry
- Return codes IBM Workload Scheduler for z/OS expects when the exit returns control to IBM Workload Scheduler for z/OS
- The function that the installation exit performs
- The processing that is performed before or after the exit.

Migration strategies

You need to consider these points when deciding the appropriate migration strategy for your enterprise:

- JES and SMF exits
- Migrating to existing subsystem definitions
- Migrating to new subsystem definitions
- Installation and verification
- Parallel testing

JES and SMF exits

JES and SMF exits supplied with IBM Workload Scheduler for z/OS can also track work for previous releases. The exits are **always downward compatible**.

Consider installing JES and SMF exits in your current production environment at least a week before you plan to migrate any of the address spaces to IBM Workload Scheduler for z/OS.

Migrating to existing subsystem definitions

You can migrate from or fall back to your current subsystems **without having to IPL the z/OS system**.

By continuing to use your current subsystem names, you do not need to consider the effect of migration on these users of IBM Workload Scheduler for z/OS services:

- Host dialog users
- PIF programs
- API programs
- Callable services
- Dynamic Workload Console

Keeping the same subsystem names reduces the installation effort of a new level of IBM Workload Scheduler for z/OS.

Migrating to new subsystem definitions

If you want to parallel-test the new level of IBM Workload Scheduler for z/OS with your current level, you must create new subsystems for the IBM Workload Scheduler for z/OS address space.

Getting the right software parts

The IBM Workload Scheduler for z/OS load modules, panels, messages, and other software parts have the same name as they had in previous IBM Workload Scheduler for z/OS releases. You must ensure that the users of IBM Workload Scheduler for z/OS services run the same level of software as the subsystem address space.

Load modules:

You can decide if you want to use the same data set name for the IBM Workload Scheduler for z/OS load modules as your previous environment. However, consider the additional effort required on your part to coordinate the JCL changes required for callers of IBM Workload Scheduler for z/OS services such as:

```
EQQEVPGM
EQQUSINx
EQQYCOM
EQQYLTOP
```

If the IBM Workload Scheduler for z/OS load library is not referenced in the STEPLIB DD statement, ensure that the IBM Workload Scheduler for z/OS library is listed first in the LNKLST concatenation **and** that the library remains empty until you are ready to cut over to IBM Workload Scheduler for z/OS on the production system. Then you should copy the load modules into the library and perform an LLA refresh.

Two IBM Workload Scheduler for z/OS load modules **must always** be in a LNKLST library:

```
EQQINITn
```

The subsystem initialization module

```
EQQSSCMn
```

The subsystem communication module.

However, this does not mean that you must reinitialize the subsystem to cut over IBM Workload Scheduler for z/OS to production. The module names defined for EQQINIT and EQQSSCM in the SYS1.PARMLIB subsystem name table (IEFSSNnn) can be overridden when an IBM Workload Scheduler for z/OS address space is created.

The EQQMINOx load module requires special attention. EQQMINOx is the scheduler's dialog interface module, is invoked by TSO SERVICES, and passes dialog requests and data to the controller. EQQMINOx must run APF authorized, therefore it must reside in an authorized library. By this token, keep in mind that any unauthorized library in a STEPLIB or LIBDEF concatenation makes the entire concatenation unauthorized. So remember to identify the library where EQQMINOx resides.

The BUILDSSX keyword of the OPCOPTS initialization statement can be used to rebuild the environment created during subsystem initialization at the new software level. When the address space is terminated, the previous environment is reinstated, thereby ensuring fallback to a previous release of IBM Workload Scheduler for z/OS.

SSCMNAME keyword of the OPCOPTS initialization statement can be used to permanently, or temporarily, replace the EQQSSCMn module that was loaded into common storage at IPL. When the TEMPORARY value is defined, the named module is loaded into private storage of the IBM Workload Scheduler for z/OS address space, therefore events created while the address space is not active will use the EQQSSCMn from the previous IPL. When PERMANENT is specified, the old EQQSSCMn in common storage is replaced.

Note:

- Create backup copies of the old load module library before you replace the objects.

The ISPF environment:

IBM Workload Scheduler for z/OS ISPF dialog users must run software parts that are at the same level as the controller address space. Again, using the same data set names for software parts libraries, such as messages and panels, negates the requirement to change TSO logon procedures.

If you use the same data set names, instruct dialog users to return to the TSO READY prompt after you have replaced the software parts and before they try to communicate with an IBM Workload Scheduler for z/OS controller.

The IBM Workload Scheduler for z/OS ISPF profile is automatically reinitialized when the EQQOPCAP panel (the IBM Workload Scheduler for z/OS main menu) is first displayed for a new release. Dialog users must enter the IBM Workload Scheduler for z/OS options dialog and redefine required values, such as the subsystem name.

Be sure to create backup copies of the old libraries before you replace the objects.

When migrating from one release of IBM Workload Scheduler for z/OS to the next, the LOADLIB, PANELLIB, MSGLIB, CLIB, and SKELLIB for the right IBM Workload Scheduler for z/OS release must be invoked from the TSO ISPF dialogs. Remember to identify the library where EQQMINOx resides.

Migration overview

This section summarizes the steps you must perform before installing a new release of IBM Workload Scheduler for z/OS. You should plan for the migration by installing and stabilizing the new IBM Workload Scheduler for z/OS release without incorporating the new functions provided. Installing a new IBM Workload Scheduler for z/OS release without initially exploiting new functions allows you to create a stable environment.

Migration steps overview

To install and activate IBM Workload Scheduler for z/OS, you must perform the following steps:

1. Ensure you have the required environment for IBM Workload Scheduler for z/OS
2. Make the necessary modifications
3. Stop the scheduler version you are currently running
4. Convert the data sets
5. Start IBM Workload Scheduler for z/OS

Only summary information appears in this section. Detailed instructions about how to make specific changes are in the chapters that follow or in referenced publications.

Establishing the required environment

You use SMP/E to install the IBM Workload Scheduler for z/OS software. Refer to the *Program Directory* for specific instructions about using SMP/E to install IBM Workload Scheduler for z/OS.

This version of IBM Workload Scheduler for z/OS must run on z/OS Version 1.13, or later.

Program requirements

Before installing IBM Workload Scheduler for z/OS, check the preventive service bucket for a current list of the required products, their maintenance levels, and recommendations from the service organizations.

The PSP for this release can be found in the TWSZOS930 upgrade. Read this document carefully before you start to install IBM Workload Scheduler for z/OS.

Your installation must have at least the earliest release supported for migration, which is IBM Workload Scheduler for z/OS Version 8.5.

Installation and verification

If you are migrating to existing subsystem definitions, you must perform these installation tasks:

1. Load the tracker software (“Step 1. Loading tracker software” on page 55).
2. Load the controller software (“Step 2. Loading controller software” on page 55).
3. Load the NLS software (“Step 3. Loading national language support software” on page 56).
4. Run the EQQJOBS CLIST (“Step 4. Using the EQQJOBS installation aid” on page 56).
5. Install JES and SMF exits at IBM Workload Scheduler for z/OS level (“Step 5. Adding SMF and JES exits for event tracking” on page 81).
6. Update PARMLIB (“Step 6. Updating SYS1.PARMLIB” on page 83).

7. Import the new default security certificates for HTTP connections (“Step 8. Securing communications” on page 99).
8. Allocate VSAM and non-VSAM data sets (“Step 9. Allocating data sets” on page 102).
9. Update the JCL procedure for the IBM Workload Scheduler for z/OS address space (“Step 10. Creating JCL procedures for address spaces” on page 128).
10. Update the initialization statements (“Step 11. Defining the initialization statements” on page 134).
11. Update the ISPF environment (“The ISPF environment” on page 188).

Ensure that you follow the subsystem verification procedures outlined in Chapter 5, “Verifying your installation,” on page 161.

You can use the conversion program for both migration and fallback. You should consider testing your installation by migrating in the production environment and then falling back.

Note: Verify that all the IBM Workload Scheduler for z/OS parameters defined in the previous release are still valid in the current release.

Parallel testing

If you want to perform the migration and then continue immediately into parallel testing in a job-tracking environment, you can use the following procedure as a guide. However, you should carefully consider the applicability of this procedure in your own IBM Workload Scheduler for z/OS configuration.

1. Stop your production system.
2. Perform data set conversion and copying.
3. Start your production system.
4. Start IBM Workload Scheduler for z/OS, Version 9.3.

You should also consider:

- If you start the JCC in both the production system and IBM Workload Scheduler for z/OS, the two JCCs cannot delete or requeue SYSOUT from the same SYSOUT class.
- Do not specify HOLDJOB(YES) or HOLDJOB(USER) for more than one of the two systems. If you do, one system might incorrectly release jobs that are held by the other system.
- When you convert the VSAM data sets, it is recommended that you run the conversion of the JS file to verify that conversion has been done correctly. Then, before running the parallel test, reallocate empty JS files. (Otherwise, the test system might find valid production JCL on the active JS file and submit it to the JES subsystem.)
- You should start with an empty JCL library data set (EQQJBLIB). Otherwise, the test system might submit production JCL incorrectly. To test that IBM Workload Scheduler for z/OS submits jobs correctly, you should create test applications with job names that are not known to the production system. JCL for those jobs could then safely be inserted into EQQJBLIB.
- On the test system you should specify TRACK(ALL) and SUBFAILURE(R) on the JTOPTS initialization statement.
- TSO commands or subroutines that have a specific name for the subsystem parameter will not report events to the test system. You should update any procedures, which are dependent on TSO commands or subroutines, if events should also be reported to the test system.

- If you are migrating from a previous release of IBM Workload Scheduler for z/OS and you use NetView or a similar product to intercept messages, make sure that WTO (write-to-operator) messages are not issued by the test system. Otherwise, the test system might trigger some processing that impacts the production system. You should not use alert WTOs, deadline WTOs, or WTO operations on the test system.
- If you want to use Restart and Cleanup when in the old subsystem is running a JCC task, you must specify the keyword DSTCLASS in the RCLOPTS statement of the new controller. The class specified in DSTCLASS must not be one of the classes specified in the JCC parameter CHKCLASS. This will prevent the JCC task from deleting the duplicate SYSOUT copy created for the Data Store before it has been successfully stored. Refer to *Customization and Tuning* for further details.

Using the preceding notes as a guide, you will be able to run one production system and one IBM Workload Scheduler for z/OS test system in parallel. The work with the database dialogs and the long-term-planning functions can be fully tested this way. The job-tracking functions of the test system will be incomplete because only the specially created test jobs will be submitted by the test system. However, the tracking of work, including the tracking of applications and jobs in the production area, will be done normally.

Migrating an end-to-end network

All the considerations for an IBM Workload Scheduler master domain manager apply to the controller. For the migration path and compatibility information for an end-to-end network, see the *Release Notes* at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

Migrating DB2 history

To migrate history data that is already defined, you need to run only the EQQICNVH sample, as explained in “Sample to migrate the history database” on page 135. Otherwise, if you want to create a new DB2 database, tables, and indexes, you need to run only the EQQINIDB sample, as explained in “Step 12. Creating the DB2 database” on page 134.

Migrating DB2 reporting

To migrate reporting data that is already defined, according to the environment where you are using DB2, perform the following actions.

DB2 for z/OS environment

Run the EQQMIGRE sample (placed in the SEQQSAMP library) to migrate the tables and views.

DB2 for distributed environment (Windows, UNIX, Linux)

If you are migrating from IBM Workload Scheduler for z/OS V9.1 or V9.2, run the following scripts. (You find the scripts on the installation CD, as described in the *Memo to Users*).

```
dbmigrate_91.bat (for Windows)
dbmigrate_91.sh (for UNIX and Linux)
```

In all other cases, run the following scripts:

```
dbmigrate.bat (for Windows)
dbmigrate.sh (for UNIX and Linux)
```

These scripts update the JHR_JOB_HISTORY_RUNS and JOS_JOB_STATISTICS tables by adding new columns. The related views are also updated with the corresponding changes.

Migrating the backup controller

After you migrated the primary controller, you can migrate the backup controller by completing the following procedure:

1. Allocate the VSAM clusters and non-VSAM data sets by running EQQPCS01 and EQQPCS02 jobs, respectively.
2. When the backup controller starts and establishes the connection with the primary controller, the primary controller sends all the required plans.
3. As soon as the backup controller receives the plans, it performs the corresponding restore.
4. To check that the restore process for all the plans has completed, look for the following message in EQQMLOG:

```
EQQN134I RE-SYNCHRONIZATION PLANS RESTORE COMPLETED
```

Changing a shared DASD tracker-to-controller connection to an NCF, XCF, or TCP/IP connection

To change a shared DASD tracker connection to an NCF (VTAM), XCF (SYSPLEX), or TCP/IP one, perform the following steps :

1. To remove the DASD connection:
 - a. In the controller started task procedure:
 - 1) Remove the EQQVDnn DD statement pointing to the event data set of the specific tracker.
 - 2) Remove the DD statement pointing to the Submit/Release data set of the tracker. Not every DASD-connected tracker has a Submit/Release data set, but if one exists, its DDNAME in the controller procedure is the same as the destination listed under the DASD keyword in the ROUTOPTS initialization statement of the controller.
 - b. In the controller initialization parameters:
 - 1) Decrease the value of the OPCOPTS ERDRTASK() keyword by the number of DASD-connected trackers being removed. If there are no DASD-connected trackers, ERDRTASK() is 0.
 - 2) Remove from the ERDRPARM() keyword the name of the PARMLIB member containing the parameters for the Event Reader task being deleted.
 - 3) Remove from the DASD keyword in the ROUTOPTS initialization statement the DDNAME of the Submit/Release data set of the tracker.
 - c. In the controller ISPF dialogs:
 - 1) Remove the DDNAME of the Submit/Release data set of the tracker from the workstation destination under dialog option 1.1.2, using the M (modify) row command.
 - 2) Remove the workstation destination from ROUTOPTS and from the workstation definition.
 - d. In the tracker started task procedure, remove the EQQSUDS DD statement.
 - e. In the tracker initialization parameters:
 - 1) In the EWTROPTS statement, set the SUREL() keyword to NO .
 - 2) In the TRROPTS statement, remove the HOSTCON(DASD) keyword.
2. To add an NCF connection:

- a. Define the VTAM LUs for the controller and the tracker. If necessary, create cross-domain definitions. IBM Workload Scheduler for z/OS requires that the LU name be the same as the ACBNAME in the APPL. For details, refer to “Step 15. Activating the network communication function” on page 143.
 - b. In the controller initialization parameters:
 - 1) In the OPCOPTS statement, set the NCFTASK() keyword to YES and write the LU name of the controller in the NCFAPPL() keyword.
 - 2) In the ROUTOPTS statement, write the LU name of the tracker in the SNA() keyword.
 - c. In the controller ISPF dialogs, write the LU name of the tracker in the workstation destination under dialog option 1.1.2, using the M (modify) row command.
 - d. In the tracker initialization parameters:
 - 1) In the OPCOPTS statement, set the NCFTASK() keyword to YES and write the LU name of the tracker in the NCFAPPL() keyword.
 - 2) In the TRROPTS statement, set the HOSTCON() keyword to SNA and write the LU name of the controller in the SNAHOST() keyword.
 - 3) In the EWTRROPTS statement, set the EWSEQNO() keyword to 1.
3. To add an XCF connection:
- a. In the SYS1.PARMLIB(COUPLEnn) member:
 - 1) Define the IBM Workload Scheduler for z/OS XCF transport class as described in “Updating XCF initialization options” on page 89.
 - 2) Define the XCF group that is to enable the controller to communicate with the trackers.
 - b. In the controller initialization parameters:
 - 1) In the XCFOPTS statement, code the GROUP(), MEMBER(), and TAKEOVER() keywords.
 - 2) In the ROUTOPTS statement, write the XCF MEMBERNAME of the tracker in the XCF() keyword.
 - c. In the controller ISPF dialogs, write the XCF MEMBERNAME of the tracker in the workstation destination under dialog option 1.1.2, using the M (modify) row command.
 - d. In the tracker initialization parameters:
 - 1) In the XCFOPTS statement, code the GROUP() and MEMBER() keywords.
 - 2) In the TRROPTS statement, set the HOSTCON() keyword to XCF.
 - 3) In the EWTRROPTS statement, set the EWSEQNO() keyword to 1.
4. To add a TCP/IP connection:
- a. Define the IP addresses for the controller and tracker. For details, refer to “Step 16. Using TCP/IP for communication” on page 146.
 - b. In the controller initialization parameters:
 - 1) In the TCPOPTS statement, set the values to define the details of the local controller. This statement is optional; if you do not specify it, the default values are taken.
 - 2) In the ROUTOPTS statement, write the TCP/IP destination name and IP address of the remote tracker in the TCPIP keyword.
 - c. In the controller ISPF dialogs, write the TCP/IP destination name of the tracker in the workstation destination under dialog option 1.1.2, using the M (modify) row command.
 - d. In the tracker initialization parameters:

- 1) In the TCPOPTS statement, set the values to define the details of the local tracker or leave the default values.
 - 2) In the TRROPTS statement, set the HOSTCON() keyword to TCP/IP and write the IP address of the controller in the TCPHOSTNAME() keyword.
 - 3) In the EWTRROPTS statement, set the EWSEQNO() keyword to 1.
5. Stop and restart the controller and tracker for the parmlib changes to take effect, and run the CP extend or the CP replan command to update the current plan with the changed workstation destinations.

Running on upgraded operating systems

To run the scheduler on a new version of the z/OS operating system, you must reassemble the SMF and JES exits mentioned in “Step 5. Adding SMF and JES exits for event tracking” on page 81 with the libraries of the new operating system.

If you upgrade the SMP/E environment to a later version of z/OS by using the SMP/E function BUILD MCS, the relink occurs automatically (ensure that the DDDEF entries for the new operating system are set up correctly by specifying the latest SEZACMTX and SCEELKED libraries). If you do not use BUILD MCS, relink the load modules by using the SMP/E function LINK LMODSCALLLIBS. With this function, all the IBM Workload Scheduler for z/OS modules are relinked to the latest SEZACMTX and SCEELKED libraries.

Migrating actions

This chapter describes the tasks you must perform to complete the migration process.

To migrate the controller by using the job stream provided with the product, see “Migrating the controller with the IWSZSELFUPGRADE application” on page 203.

If after migrating you need to return to the previous version, see “Performing fallback” on page 207.

Migrating data sets

For migration purposes, data sets fall into three categories:

- VSAM data sets that are copied and converted by the EQQICTOP migration program
- Non-VSAM data set that you can copy, or use unchanged, in the new version
- VSAM and non-VSAM data sets that must be empty when you migrate to IBM Workload Scheduler for z/OS

Each of these categories is described in the following sections.

EQQICTOP VSAM data set conversion program

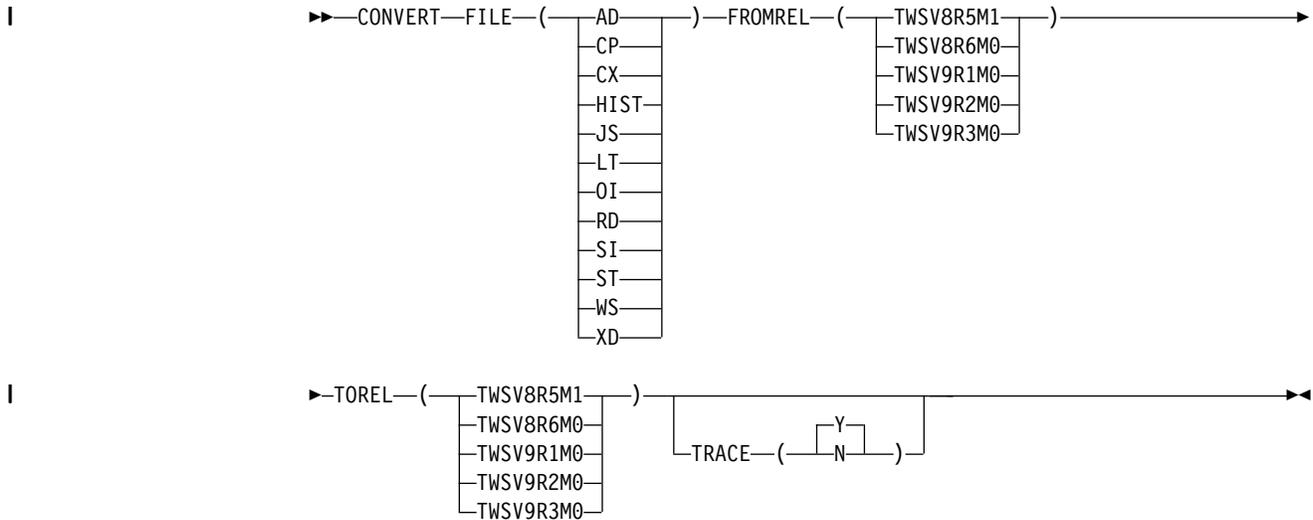
Purpose

With the EQQICTOP conversion program, you can migrate VSAM data sets from earlier releases of IBM Workload Scheduler for z/OS. You can also use the program to reverse the procedure in case you need to fall back to your old system.

The EQQJOBS program creates JCL tailored to your installation specifications in the EQQICNVS and EQQICNVH members.

EQQICTOP is controlled by CONVERT statements in the SYSIN file. You can supply any number of these statements to EQQICTOP.

Syntax



Parameters

FILE(*file identifier*)

Defines the data set to be converted. You can specify one of the following file identifiers on each CONVERT statement:

- AD** Application descriptions and JCL variable tables
- CP** The current plans, EQQCPnDS and EQQNCPDS
- CX** The current plan extension, EQQCXDS and EQQNCXDS
- HIST** DB2 operation history data from one release to another
- JS** JCL repository and retrieved job logs
- LT** Long-term plan
- OI** Operator instructions
- RD** Special resource definitions
- SI** Side information file, ETT criteria and configuration information
- ST** Step awareness (only if the Step Awareness feature is active on the primary controller)
- WS** Workstation descriptions, calendars, and periods
- XD** Extended dependencies

Your conversion JCL should contain DD names EQQxxIN and EQQxxOUT for each data set that you want to convert, where xx is the file identifier.

FROMREL(*product identifier*)

Defines the product and release level of the input data set. You can specify one of the following:

TWSV8R5M1

IBM Workload Scheduler for z/OS Version 8 Release 5
Modification Level 1

TWSV8R6M0

IBM Workload Scheduler for z/OS Version 8 Release 6

|
| **TWSV9R1M0**

| IBM Workload Scheduler for z/OS Version 9 Release 1

| **TWSV9R2M0**

| IBM Workload Scheduler for z/OS Version 9 Release 2

| **TWSV9R3M0**

| IBM Workload Scheduler for z/OS Version 9 Release 3

| **TOREL**(*product identifier*)

Defines the product and release level of the output data set. You can specify one of the following:

| **TWSV8R5M1**

| IBM Workload Scheduler for z/OS Version 8 Release 5
| Modification Level 1

| **TWSV8R6M0**

| IBM Workload Scheduler for z/OS Version 8 Release 6

| **TWSV9R1M0**

| IBM Workload Scheduler for z/OS Version 9 Release 1

| **TWSV9R2M0**

| IBM Workload Scheduler for z/OS Version 9 Release 2

| **TWSV9R3M0**

| IBM Workload Scheduler for z/OS Version 9 Release 3

| **Note:**

1. Conversion stops if there is a VSAM I/O error on one of the files. One such error is a duplicate key on the output file. This can occur if the output data set is not empty.
2. Migrate the currently active JCL-repository data set. You can check whether the primary or alternate data set is in use by selecting option 6 in the Query Current Plan dialog. Do this when no work is running and before you stop the controller.
3. You can use one of two methods to convert the current plan:
 - If the last action performed on your production system was to extend or replan the current plan, use the new-current-plan data set that was created on this system as input to the conversion program. This is the preferred method as it ensures you will not lose any job-tracking records, this is relevant if you use the track log (EQQTROUT) as an audit trail.
 - If no error occurred when you stopped your production system, both primary and alternate current plans are the same. Use EQQCP1DS as input to the conversion program.

In both cases, the output file **must** be the new-current-plan data set (EQQNCPDS) on your IBM Workload Scheduler for z/OS system.

You can convert the current plan extension data set using the same methods. If the EQQCPIN DD card in the EQQICNVS conversion program refers to EQQNCPDS, the EQQCXIN DD card must refer to the corresponding EQQNCXDS created with the latest REPLAN or EXTEND job. If, instead, no REPLAN or EXTEND job was run before shutting down the system, the EQQCXIN DD card must refer to the latest EQQCXDS data set in use. In both cases, the EQQICNVS output file for CX must be the new-current-plan extension data set (EQQNCXDS) corresponding to the new-current-plan data set (EQQNCPDS).

4. In addition to input and output DD names for each VSAM file, the migration JCL should also contain the EQQMLOG and EQQMLIB DD

names. EQQMLOG is an output file for messages. EQQMLIB is an input file that contains the product message library.

TRACE(Y|N)

Specifies if message:

```
EQQIC66I PROCESSING APPLICATION AD_data_set_name VALID FROM From_date
STATUS status
```

is to be issued in the message log of the migration job as each data set of the Application Description database is migrated to the new product release.

Set TRACE to N specifically to inhibit the issue of message EQQIC66I; otherwise, the message is written to the MLOG as part of the migration process.

Note: Specifying TRACE(Y), or not specifying it at all (the default), will cause one occurrence of message EQQIC66I to be issued for each processed application. Consider using TRACE(N) if you do not want this message to be issued in the batch output MLOG.

Example:

```
| //OPCMIG JOB (777777,777),'Migrate to Tivoli Workload Scheduler for z/OS V9R2M0,
| // MSGLEVEL=(1,1), NOTIFY=&SYSUID,MSGCLASS=H,CLASS=A
| // *
| //CONVERT EXEC PGM=EQQICTOP,REGION=2048K
| //STEPLIB DD DISP=SHR,DSN=OPC.INST.LOADLIB
| //EQQMLIB DD DISP=SHR,DSN=OPC.INST.SEQMSG0
| //EQQMLOG DD SYSOUT=*
| //EQQADIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.AD
| //EQQADOUT DD DISP=OLD,DSN=CCOPC.OPCC.AD
| //EQQWSIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.WS
| //EQQWSOUT DD DISP=OLD,DSN=CCOPC.OPCC.WS
| //EQQCPIIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.NCP
| //EQQCPIOUT DD DISP=OLD,DSN=CCOPC.OPCC.NCP
| //EQQCXIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.NCX
| //EQQCXOUT DD DISP=OLD,DSN=CCOPC.OPCC.NCX
| //EQQLTIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.LT
| //EQQLTOUT DD DISP=OLD,DSN=CCOPC.OPCC.LT
| //EQQJSIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.JS1
| //EQQJSOUT DD DISP=OLD,DSN=CCOPC.OPCC.JS1
| //EQQOIIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.OI
| //EQQOIOUT DD DISP=OLD,DSN=CCOPC.OPCC.OI
| //EQQSIIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.SI
| //EQQSIOUT DD DISP=OLD,DSN=CCOPC.OPCC.SI
| //EQQSTIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.ST
| //EQQSTOUT DD DISP=OLD,DSN=CCOPC.OPCC.ST
| //EQQRDIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.RD
| //EQQRDOUT DD DISP=OLD,DSN=CCOPC.OPCC.RD
| //EQQXDIN DD DISP=SHR,DSN=CCOPC.OPCC.OLD.NXD
| //EQQXDOUT DD DISP=OLD,DSN=CCOPC.OPCC.NXD
| //SYSIN DD *
|
| /* MIGRATION FROM Tivoli Workload Scheduler for z/OS V9.2.0 to */
| /* IBM Workload Scheduler for z/OS V9.3.0 IS ASSUMED */
| CONVERT FILE(AD) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0) TRACE(Y)
| CONVERT FILE(CP) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(CX) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(WC) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(LT) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(JS) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(OI) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
| CONVERT FILE(RD) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
```

```

CONVERT FILE(SI) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
CONVERT FILE(ST) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)
CONVERT FILE(XD) FROMREL(TWSV9R2M0) TOREL(TWSV9R3M0)

```

In this example, all VSAM files are converted from a previous release to IBM Workload Scheduler for z/OS format. The tasks performed immediately before this job was submitted are listed here in order:

1. Verified JS1 as the active JCL repository in option 6.6 on the previous controller. If JS2 is the active JCL repository, use that as input but be sure to use JS1 as output because IBM Workload Scheduler for z/OS by default uses JS1 as the active JCL repository when you start a subsystem with an empty checkpoint data set.
2. The previous controller was shut down normally, as verified in the message log. Check that a current plan backup process was completed after the stop command was received by the subsystem.
3. A batch job was submitted to allocate and back up the previous data sets to new DSNs.
4. EQQPCS01 from IBM Workload Scheduler for z/OS EQQJOBS was submitted to allocate the VSAM clusters required for IBM Workload Scheduler for z/OS.
5. The old NCP is used as input if a daily plan batch process was submitted on the previous system prior to shut down. Output is the IBM Workload Scheduler for z/OS NCP.

Data sets that you need to migrate

Allocate new VSAM data sets for IBM Workload Scheduler for z/OS. Existing data can then be migrated by using EQQICTOP. Keep a copy of the old data sets for backup and fallback purposes. The following data sets must be migrated to IBM Workload Scheduler for z/OS format:

Table 34. Data sets that you need to migrate

DD Name	Description
EQQADDS	Application descriptions and JCL variable tables
EQQJSnDS	JCL repository (currently active)
EQQLTDS	Long-term plan
EQQNCPDS, or EQQCPnDS	The current plan, but use the NCP as input if a daily plan process created an NCP after the old system was shut down.
EQQNCXDS, or EQQCXnDS	The current plan extension, but use the NCX as input if a daily plan process created an NCX after the old system was shut down.
EQQOIDS	Operator instructions
EQQRDDS	Special resource definitions
EQQSIDS	Side information, ETT criteria and configuration information
EQQSTDS	Step Awareness (only if the feature is active on the primary controller)
EQQWSDS	Workstation descriptions, calendars, and periods

Data sets that can be used without migrating

IBM Workload Scheduler for z/OS can use unchanged data from the following data sets:

Table 35. Data sets that IBM Workload Scheduler for z/OS can use

DD Name	Description
EQQEVLIB	Configuration file repository for event-triggered resource handling
EQQINCWK	JCC incident work file
EQQJBLIB ¹	JCL library
EQQJCLIB	Job-completion-checker (JCC) message-table library
EQQJTABL	Job table log file
EQQOUCV	Data set used for controller-output collector communication
EQQPRLIB	Automatic-recovery-procedure library
EQQSCLIB	Script library for end-to-end scheduling with fault tolerance capabilities
—	JCC incident log
¹ If this library contains jobs for the scheduler planning, the JCL must be modified to reflect the new installation.	

Empty data sets

When you have completed your testing of IBM Workload Scheduler for z/OS and have performed data set migration, ensure that the following data sets are empty before you start the product for the first time in production:

EQQACPDS

Archive of old current plan

EQQBKPT

Backup checkpoint

EQQCKPT

Checkpoint

EQQCXDS

Current plan extension

EQQDL^{mm}

Dual job-tracking logs

EQQEVDS

Event data sets

EQQHTTP0

Event data set for end-to-end scheduling with z-centric capabilities

EQQJTARC

Job-tracking archive

EQQJT^{mm}

Job-tracking logs

EQQLOGRC

Job log and Restart Information pending requests Log data set

EQQMLOG

Message log

EQQMONDS

Monitoring Task Data Set

EQQNSTD

New Step Awareness

EQQSCPDS

Secondary Current Plan Data Set

EQQSTC

Started-task submit

EQQSUDS

Submit release

EQQTROUT

Job-tracking log copy created by the daily planning jobs. Input to the EQQAUDIT program, that is not downward compatible

EQQTWSIN

Input event data set for end-to-end scheduling with fault tolerance capabilities

EQQTWSOU

Output event data set for end-to-end scheduling with fault tolerance capabilities

EQQTWSCS

Centralized script work repository

EQQCP n DS , EQQCXDS, and EQQSCPDS do not have to be empty, but they can be. When the product is started for the first time, you **must** specify CURRPLAN(NEW) on the JTOPTS statement. Therefore, any data in the EQQCP n DS and EQQCXDS data sets will immediately be replaced by the contents of EQQNCPDS and EQQNCXDS. Similarly, the inactive EQQJS n DS data set (EQQJS2DS in this example) does not have to be empty, although it can be.

Tracker, Data Store, and Output Collector considerations

When you migrate a tracker, it is not necessary for EQQEVDS and EQQSUDS to be empty. The migration enables you to use the subsystem with the new release and modifies the JCL trackers to point to the libraries used, for example EQQMLIB or STEPLIB.

When you migrate a Data Store, consider that:

- In the Data Store, it is not necessary for EQQPKI01, EQQSKI01, EQQSDF nm , EQQUDF nm to be empty.
- In the controller:
 - If the CP files are kept, it is not necessary for EQQPKI01, EQQSKI01, EQQSDF nm to be empty.
 - If the CP files are deleted, EQQPKI01, EQQSKI01, EQQSDF nm must be empty.

The data store and controller tasks might be migrated at different times, provided that the maintenance level of the old and new release of IBM Workload Scheduler for z/OS is the same. This means that you should apply any PTF which affects both controller and data store code to both releases of the product. If this level of concurrent PTF maintenance cannot be maintained, it is best to keep the data store and controller on the same release of IBM Workload Scheduler for z/OS. If the migration is successfully performed, you should be able to use the Restart and Cleanup function on the new release for any operation which was on the error list on the old release of IBM Workload Scheduler for z/OS

If you change a datastore connection type and you want to reflect the naming convention in the FLOPTS destination name, keep the former destination name in the FLOPTS parameter that corresponds to the connection type to be used (SNADEST, XCFDEST, or TCPDEST). For example, suppose that you change the datastore connection from SNA to XCF and the former FLOPTS is SNADEST(OPCTRK1.DST). If you want to use XCFTRK1.DST as new destination name, specify the following FLOPTS parameter: XCFDEST(OPCTRK1.DST, XCFTRK1.DST). Omitting the former destination produces the messages EQQFL18E and EQQM643W in the controller message log, when retrieving any joblog stored with the former destination name.

Output collector is not to be migrated.

Migrating the production environment

To migrate your production system, perform the following steps:

1. "Close down your production system"
2. "Convert VSAM files to the new system format"
3. "Initialize the new system" on page 202
4. "Produce a checkpoint data set containing data from the old production system" on page 202
5. "Start the new system" on page 202
6. "Validate the new system" on page 203

Close down your production system

1. From the Service Functions dialog on the production system (OPCA), deactivate job submission for jobs running on fault-tolerant workstations.
2. Unlink all fault-tolerant workstations in the network by using one of the available IBM Workload Scheduler interfaces.
3. Before you proceed with the next steps, wait until all the events are processed in the EQQTWSIN and EQQTWSOU data sets. To verify this, use the sample utility EQQCHKEV, provided in the sample library.

The EQQCHKEV utility checks the data set structure of EQQTWSIN and EQQTWSOU which are the input and output end-to-end event data sets from the version you are migrating. The utility provides an informational message indicating the number of events still to be processed. When the data set contains zero unprocessed events you can proceed with the migration. The utility also checks the integrity of the data sets and issues an appropriate error message in case of corruption or inconsistency.
4. From the Daily Plan dialog on the production system (OPCA), create a replan or plan-extend batch job. Change the job card to contain TYPRUN=HOLD, and submit the job. Save the JCL in a data set in case you have to resubmit it to correct an error.
5. If you specified CHECKSUBSYS(YES) on the BATCHOPT statement used by the batch job, change it to CHECKSUBSYS(NO). In the BATCHOPT statement used by the batch job, comment out the TPLGYPRM and JRUNHISTORY keywords, if they are used.
6. Using the Query Current Plan dialog on the production system (OPCA), check which JS file is currently in use on this system.
7. Stop OPCA and OPCS, release the daily plan from hold and make sure it runs successfully.

Convert VSAM files to the new system format

1. Create a backup copy of the IBM Workload Scheduler for z/OS VSAM files.
2. Allocate VSAM and non-VSAM data sets for IBM Workload Scheduler for z/OS by using the EQQPCS01 and EQQPCS02 jobs.
3. Review the EQQICNVS sample job. Ensure that input and output data set names are correctly specified. Make sure you select the current JS file. When defining input and output files for the CP file conversion, use the NCP file, because a new current plan has just been created.
4. Run EQQICNVS to convert the VSAM data to IBM Workload Scheduler for z/OS format.
5. Verify that the conversion program ran successfully. If there are any problems converting the VSAM files, you should abandon the migration.
6. Back up the IBM Workload Scheduler for z/OS non-VSAM data sets.

7. If you have stopped migrating, start OPCA, OPCB, and OPCC. Release any held queues and restart any drained initiators.

Initialize the new system

Before you perform the steps described in this section, ensure that the VSAM file conversion described in the preceding section was successful.

1. Ensure that the data sets referred to in “Empty data sets” on page 199 are empty. Use ISPF browse to ensure that all job-tracking logs (EQQJTnn), the job-tracking archive (EQQJTARC), and the checkpoint (EQQCKPT) data sets are empty. If you use dual job-tracking logs (EQQDLnn), they should also be empty.
2. Modify the JCL procedure for OPCA to include the new DD names and data sets added in IBM Workload Scheduler for z/OS.
3. Modify initialization parameters for OPCA. The CKPT data set is not yet initialized the first time you start OPCA after migration, so you must specify CURRPLAN(NEW) in JTOPTS. Specify BUILDSGX(REBUILD) and SSCMNAME(EQQSSCMM,TEMPORARY) in the OPCOPTS initialization statement. Specify the PIFHD keyword in the INTFOPTS initialization statement. As soon as OPCA has started, change back to CURRPLAN(CURRENT), to prevent OPCA from recovering from the new current plan each time it starts.

Note: You might find it useful to specify JOBSUBMIT(NO) and FTWJSUB(NO) in the JTOPTS initialization statement so that work is not submitted when you start OPCA. When you have checked that OPCA has started without errors, you can activate job submission using the Service Functions dialog.

To initialize the checkpoint data set, specify OPCHOST(YES) in OPCOPTS. This is so that, when the scheduler starts, the NMM task initializes the checkpoint data set with FMID and LEVEL corresponding to SSX. The OPCHOST value can then be changed. For example, you can change the value to OPCHOST(PLEX) when the subsystem is used as the controlling system in XCF.

4. Run the EQQPCS05 and EQQPCS06 jobs to create the work directory.
5. Start OPCA. Verify that no errors occurred during initialization. If required, correct any errors and restart OPCA.
6. Stop OPCA.

Produce a checkpoint data set containing data from the old production system

Produce a checkpoint data set containing data from the old production system:

1. Merge OLD.CKPT, from the version you are migrating, and the newly allocated CKPT, created in the previous section, into CKPT.NEW using sample EQQPMCKP, which you customize for your environment.
2. Back up the current CKPT and then rename CKPT.NEW to the current CKPT.

Start the new system

1. Change JTOPTS CURRPLAN(NEW) to CURRPLAN(CURRENT).
2. In the BATCHOPT statement used by the batch job, uncomment the TPLGYPRM and JRUNHISTORY keywords if you had commented them out.
3. Start the controller OPCA. The merged checkpoint data set will enable it to continue reading the event records.
4. Start all the trackers without BUILDSGX. Ensure that the load modules invoked are still those for the version from which you are migrating.

5. Stop the trackers after the events in CSA are processed.
6. Modify initialization parameters for OPCB. Specify BUILDSSX(REBUILD) and SSCMNAME(EQQSSCMM,TEMPORARY) on the OPCOPTS initialization statement. Specify the PIFHD keyword on the INTFOPTS initialization statement.
7. Start the OPCB and OPCS.
8. Enter the Service Functions dialog on OPCA, and activate job submission (if it is not already active).
9. Start the OPCC and OPCD systems.
10. Submit a daily plan replan or extend **as soon as possible** after migration. Until a new current plan is created, any references to special resources will cause the resource object to be copied from the EQQRDDS to the current-plan-extension data space. This processing has some performance overheads.
The new-current-plan-extension data set (EQQNCXDS) is built during daily planning to contain all special resources referenced by operations in the new current plan.

Validate the new system

1. From the Ready List dialog, review the status of active operations.
2. Check that the operations that are becoming ready on the workstations representing the three z/OS systems are successfully submitted to the intended system. Also check that the ending status is correctly reflected in the ready lists.
3. Verify that the current plan and the long-term plan can be extended successfully.
4. Verify that other IBM Workload Scheduler for z/OS-related processes (for example, the dialogs, batch programs, and PIF-based programs) work as expected.

Migrating the controller with the IWSZSELFUPGRADE application

You can upgrade the IBM Workload Scheduler for z/OS controller in an automatic way, with only few manual steps, by using the IWSZSELFUPGRADE application that is provided with the product.

In IWSZSELFUPGRADE, the jobs requiring manual actions are defined as dummy operations on a manual start and completion workstation.

IWSZSELFUPGRADE is provided in batch loader and Workload Automation Programming Language formats that you can import by using the EQQUPGBL or EQQUPGWA sample, respectively.

After importing the IWSZSELFUPGRADE application, to migrate an IBM Workload Scheduler for z/OS controller complete the following procedure.

Note:

1. Before running IWSZSELFUPGRADE, ensure that you set VARSUB=YES in the OPCOPTS statement.
2. If you are using JES3 exit, modify the operations 004 and 005 in IWSZSELFUPGRADE to replace EQQJES2 and EQQJES21 with EQQJES3.
1. Copy IWSZSELFUPGRADE to the Application Description.
2. Customize the following jobs as required for your migration purposes:

|
| **EQQALPDS**

| To allocate the data sets required to run EQQJOBS.

| **EQQNPJOB**

| To mark as NOP all the operations that install or migrate optional
| functions (such as Restart and Cleanup or Reporting) that you do not
| use.

| **EQQCPMOD**

| To copy the load modules EQQSSCM n and EQQINIT n to the user
| library that needs to be APF authorized and added to the z/OS system
| LINKLIST.

| **EQQCPPAR**

| To copy the old PARMLIB to the new PARMLIB.

- | 3. Add the IWSZSELFUPGRADE application to the current plan.

| The following operations are run automatically or wait for your manual
| intervention:

| **Operation 001 (automatic)**

| The data sets required for EQQJOBS are allocated.

| **Operation 002 (manual)**

| You are required to run EQQJOBS and copy the generated sample JCLs
| to the job library data set (EQQJBLIB).

| **Note:** Ensure that in the Create Sample Job JCL (EQQJOBS3) panel, you
| specify the //&0JOBNAME JOB card in the Job Statement Information
| field.

| **Operation 003 (automatic)**

| The NOPJOB job is run.

| **Operations 004, 005, and 006 (automatic)**

| The samples generated by EQQJOBS named EQQJES2, EQQJES21, and
| EQQSMF are run to link the JES2 and SMF exits.

| **Note:** If you are using JES3 exit, you must have modified the
| operations running EQQJES2 and EQQJES21 to an operation running
| EQQJES3.

| **Operation 009 (manual)**

| You are required to manually update the following members:

- IEFSSN nm defined for the load modules EQQINIT n and EQQSSCM n .
- IKJTSON n defined for the load module EQQMINO x .

| For detailed information about the load modules, see "Load modules"
| on page 187.

| **Operations from 010 to 113 (automatic)**

| The following jobs are automatically run, unless you marked them as
| NOP.

| **Note:** Because EQQPCS02 contains system symbols, if you want to use
| them you must make EQQPCS02 a started task or batch job by copying
| it to a procedure library. Then, define the workstation where this
| operation is run as STARTED TASK, STC = Y.

- EQQRCERT
- EQQPCS01

- EQQPCS02
- EQQPCS03
- EQQPCS04
- EQQPCS05
- EQQPCS06
- EQQPCS07
- EQQPCS08
- EQQPCS09
- EQQPCS10
- EQQPCS11
- EQQPCS12
- EQQPCS13

Operation 120 (manual)

You are required to update the controller startup procedure.

Operation 130 (automatic)

The EQQICNVS sample job is automatically run to migrate the VSAM data sets.

Operation 131 (automatic)

The EQQICNVH sample job is automatically run to migrate the history DB2 tables.

Operation 150 (automatic)

The COPYMOD job is automatically run to copy the load modules EQQSSCM n and EQQINIT n to the PARMLIB.

Operation 151 (automatic)

The COPYPARM job is automatically run to copy the old PARMLIB to the new PARMLIB.

Operation 152 (manual)

You are required to update the new PARMLIB as required.

After the IWSZSELFUPGRADE application completes successfully, you can start the subsystem that was migrated.

Installing or upgrading the IBM Workload Scheduler for z/OS agent automatically

You can automatically install or upgrade an IBM Workload Scheduler for z/OS agent (z-centric) or an agent with dynamic capabilities by customizing and running the following applications provided with the product. The applications are provided in both batch loader and Workload Automation Programming Language formats.

IWSZCENINSTALL

To upgrade an IBM Workload Scheduler for z/OS agent. Run this application on a workstation where an IBM Workload Scheduler for z/OS agent instance exists, to automatically upgrade it.

Use the EQQZCEBL sample to import the batch loader format or the EQQZCEWA sample to import the Workload Automation Programming Language format.

IWSZZREINSTALL

To install an IBM Workload Scheduler for z/OS agent by running a Remote Command job defined on the Dynamic Workload Console. Run this application on a workstation where an IBM Workload Scheduler for z/OS agent instance exists, to automatically install as many IBM Workload Scheduler for z/OS agents on as many workstations as you want.

Use the EQQZREBL sample to import the batch loader format or the EQQZREWA sample to import the Workload Automation Programming Language format.

IWSZZDYNINSTALL

To upgrade a dynamic agent. Run this application on a workstation where a dynamic agent instance exists, to automatically upgrade it.

Use the EQQZDYBL sample to import the batch loader format or the EQQZDYWA sample to import the Workload Automation Programming Language format.

Before importing the IWSZZCENINSTALL, IWSZZREINSTALL, or IWSZZDYNINSTALL application to the AD database, complete the following customization steps:

1. The applications apply to the Linux operating system. To use them in a Windows environment, modify the paths and commands within each sample job as required.
2. The workstation names associated with the operations within the samples are A130 and Z130. Ensure that you either create these workstations in your environment or modify the samples with your actual workstation names, as follows:

A130 The agent running the command (*driving agent*).

Z130 The agent that is to be installed (*target agent*).

3. All the jobs that are run by the IWSZZCENINSTALL, IWSZZREINSTALL, or IWSZZDYNINSTALL application are defined in the EQQZCJCL sample. Create the corresponding jobs in the controller JOBLIB and customize them according to your environment, as follows:

/opt/IBM/TWA_tws93zc

Installation path of the target agent.

Buildzcen93

Directory of the target agent where the agent will be installed.

/mnt/build

Source directory on the target agent from which the new agent code will be copied.

-uname tws93zc

Name of the user for which the IBM Workload Scheduler for z/OS agent is installed.

/JDBC_drivers

Installation path of the JDBC drivers on the target agent.

The IWSZZCENINSTALL, IWSZZREINSTALL, and IWSZZDYNINSTALL applications comprise the following jobs, connected by internal dependencies:

Operation 001, workstation A130, job name JOBSTOP

Stops the target agent.

Operation 002, workstation A130, job name JOBUNINS

Uninstalls the target agent by running the twsinst command.

| **Operation 003, workstation A130, job name JOBDLBD**

| Deletes the source directory containing the old tar file for the target
| agent.

| **Operation 004, workstation A130, job name JOBDLINS**

| Deletes the target agent installation path.

| **Operation 005, workstation A130, job name JOBCOPY**

| Copies the tar file to install the agent from the driving agent to the
| target agent installation path.

| **Operation 006, workstation A130, job name JOBUNTAR**

| Runs the untar command of the agent code.

| **Operation 007, workstation A130, job name JOBINSZC (in IWSZZCENINSTALL)
| or JOBINSYD (in IWSZZDYNINSTALL)**

| Installs the target agent by running the twsinst command.

| **Operation 008, workstation A130, job name JOBJDBC (automatic)**

| Copies the JDBC drivers to the target agent.

| **Operation 009, workstation WAIT**

| This operation waits for a few minutes, to ensure that the target agent
| is started.

| **Operation 010, workstation Z130, job name**

| Runs a sample job on the target agent to verify that the installation was
| successful.

| Choose which application you want to run and use the appropriate sample to load
| it into your AD database.

Performing fallback

If a problem occurs after IBM Workload Scheduler for z/OS has been active as a production system for some time, and the problem is serious enough, you might need to stop the new system and return the workload to the previous system. You can do this by using a procedure called *fallback*, if the IBM Workload Scheduler for z/OS data sets are usable.

| **Note:** If on the primary controller the Step Awareness feature was active and you
| want to keep it in the previous system, ensure that you convert the EQQSTDS data
| set.

The fallback procedure is as follows:

1. Run the EQQPCS01 and EQQPCS02 jobs to allocate new data sets for the old production system. The current data sets, or a copy, used by the IBM Workload Scheduler for z/OS systems should be kept for problem determination purposes.
2. Run the EQQPCS05 job to create the work directory.
3. If required, close down the systems in the same way as during migration. This is required if the current plan on the OPCA system is intact and job tracking is working normally.
4. If possible, create up-to-date data sets for the long-term plan and new current plan for the OPCA system.

Do not submit a REPLAN job prior to shutdown, unless the PERMANENT option was used for SSCMNAME on the converted system, or if SSCMNAME was not specified.

If SSCMNAME(EQQSSCMM,TEMPORARY) was used, message EQQX145E will be issued if a REPLAN job is started after the controller is shut down.

- Build VSAM data sets for the old system by running the EQQICNVS job to convert IBM Workload Scheduler for z/OS files to their previous format.

Note: Before running EQQICNVS job, check that a daily plan batch process was not submitted on the system before shutting the controller down. Then, in EQQICNVS set the IBM Workload Scheduler for z/OS CP1 and XD1 as the input, and NCP and NXD as the output.

In the following example, all VSAM files are converted from the current version of IBM Workload Scheduler for z/OS to the format of the previous release of the product.

```
//OPCBK JOB (777777,777),'Fallback to V9R2',MSGLEVEL=(1,1),
//      NOTIFY=&SYSUID,MSGCLASS=H,CLASS=A
//*
//CONVERT EXEC PGM=EQQICTOP,REGION=2048K
//STEPLIB DD DISP=SHR,DSN=OPCESA.INST.LOADLIB
//EQQMLIB DD DISP=SHR,DSN=OPCESA.INST.SEQQMSG0
//EQQMLOG DD SYSOUT=*
//EQQADIN DD DISP=SHR,DSN=CCOPC.OPCC.AD
//EQQADOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.AD
//EQQWSIN DD DISP=SHR,DSN=CCOPC.OPCC.WS
//EQQWSOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.WS
//EQQCPIN DD DISP=SHR,DSN=CCOPC.OPCC.CP1
//EQQCPOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.NCP
//EQQLTIN DD DISP=SHR,DSN=CCOPC.OPCC.LT
//EQQLTOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.LT
//EQQJSIN DD DISP=SHR,DSN=CCOPC.OPCC.JS1
//EQQJSOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.JS1
//EQQOIN DD DISP=SHR,DSN=CCOPC.OPCC.OI
//EQQOOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.OI
//EQQSIIN DD DISP=OLD,DSN=CCOPC.OPCC.SI
//EQQSIOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.SI
//EQQSTIN DD DISP=OLD,DSN=CCOPC.OPCC.ST
//EQQSTOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.ST
//EQQCXIN DD DISP=OLD,DSN=CCOPC.OPCC.CX
//EQQCXOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.NCX
//EQQRDIN DD DISP=OLD,DSN=CCOPC.OPCC.RD
//EQQRDOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.RD
//EQQXDIN DD DISP=SHR,DSN=CCOPC.OPCC.XD1
//EQQXDOUT DD DISP=OLD,DSN=CCOPC.OPCC.OLD.NXD
//SYSIN DD *
/* FALLBACK FROM IBM Workload Scheduler for z/OS V9.3.0 to V9.2.0
   IS ASSUMED */
CONVERT FILE(AD) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(CP) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(WO) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(LT) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(JS) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(OI) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(CX) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(RD) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(SI) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
CONVERT FILE(ST) FROMREL(TWSV9R3M0) TOREL(TWSV9R2M0)
```

- Ensure that in the JTOPTS initialization statement you set CURRPLAN(NEW).

Note: You might find it useful to specify JOBSUBMIT(NO) and FTWJSUB(NO) in JTOPTS, so that work is not submitted when you start OPCA. When you have checked that old system has started without errors, you can activate job submission using the Service Functions dialog.

- Start the OPCA and OPCB systems again using the converted files and start OPCS.

8. If the new current plan (NCP) data set is not fully up-to-date because you could not run the daily plan program, use the MCP dialog to update the status of operations to make the current plan up-to-date.
9. Start the OPCC and OPCD systems again. Use the SSCMNAME keyword on the JTOPTS initialization to load the current subsystem communication module for the release to which you are falling back.

Part 3. IBM Workload Scheduler for z/OS connector

This part describes how to plan, install, configure, and uninstall the IBM Workload Scheduler for z/OS connector.

Chapter 7. Installing, upgrading, and uninstalling IBM Workload Scheduler for z/OS connector on distributed systems

How to install, upgrade, and uninstall IBM Workload Scheduler for z/OS connector on distributed systems.

To use the Dynamic Workload Console you must install the IBM Workload Scheduler for z/OS connector.

IBM Workload Scheduler for z/OS connector is automatically installed with a Dynamic Workload Console instance.

It is not possible to have a stand-alone instance of an IBM Workload Scheduler for z/OS connector V9.3.

To upgrade an existing version of a z/OS connector, you have one of the following scenarios:

IBM Workload Scheduler for z/OS connector single instance

For more information about the procedure to upgrade single instance, see “Upgrading IBM Workload Scheduler for z/OS connector single instance on distributed systems.”

Multiple instance contains an IBM Workload Scheduler for z/OS connector and a Dynamic Workload Console

For more information about the procedure to upgrade multiple instance, see “Procedure to upgrade the Dynamic Workload Console and the z/OS connector” on page 219.

IBM Workload Scheduler for z/OS DVDs to use

Which IBM Workload Scheduler for z/OS DVDs to use.

About this task

Table 36. IBM Workload Scheduler for z/OS bundles

To install IBM Workload Scheduler for z/OS component:	DVD to use:
IBM Workload Scheduler for z/OS Agent (z-centric)	z-centric disk
Dynamic Workload Console with a z/OS connector	DWC disk
batch reports	DWC disk

Upgrading IBM Workload Scheduler for z/OS connector single instance on distributed systems

Upgrading IBM Workload Scheduler for z/OS connector single instance on distributed systems

You cannot upgrade the z/OS connector V8.6.0 and related fix packs to version 9.3 directly.

It is not possible to upgrade a stand-alone IBM Workload Scheduler z/OS connector installed in the <ZOSCONN_INST_DIR> directory.

You can configure the old z/OS connector instance in a new instance of z/OS connector installed with a Dynamic Workload Console V9.3 in the <DWC_NEW_INST_DIR> directory.

You must first save the configuration files in the old z/OS connector instance by using the *wastools* utilities and then import the configuration files by using the *wastools* utilities in the new Dynamic Workload Console instance.

To configure the z/OS connector in a new instance of Dynamic Workload Console, perform the following steps:

1. Install a new instance of Dynamic Workload Console in the directory <DWC_NEW_INST_DIR>, as described in “Installation procedure for Dynamic Workload Console” on page 259.
2. Export the z/OS connector configuration properties that are saved in the <ZOSCONN_INST_DIR> directory, by running:

On Windows operating systems

```
<ZOSCONN_INST_DIR>\wastools\displayZosEngine.bat
```

On UNIX and Linux operating systems

```
<ZOSCONN_INST_DIR>/wastools/displayZosEngine.sh
```

Note: If you are connected to multiple controllers, you have to repeat this step for each connection you want to maintain in the new z/OS connector configuration.

3. Import the z/OS connector configuration properties in the Dynamic Workload Console instance that is installed in the <DWC_NEW_INST_DIR>, by running:

On Windows operating systems

```
<DWC_NEW_INST_DIR>\TWSZOS\wastools\createZosEngine.bat
```

On UNIX and Linux operating systems

```
<DWC_NEW_INST_DIR>/TWSZOS/wastools/createZosEngine.sh
```

Note: If you want to maintain the connections to multiple controllers, you have to repeat this step for each connection previously defined that you want to save.

Upgrading z/OS connector installed with one or more components in the same directory

Due to IBM Workload Scheduler installation infrastructure changes described in “Upgrading overview” on page 281, to upgrade z/OS connector multiple instances installed in the same directory, you must follow the procedures listed in Table 37 on page 215.

Table 37. Upgrade deployment model for Dynamic Workload Console multiple instances installed in the same directory

z/OS connector multiple instance in the same directory contains:	Procedure to run:
Fault-tolerant agent V8.5.0, V8.5.1 or V8.6.0 and related FixPacks Dynamic Workload Console V8.5.0, V8.5.1 or V8.6.0 and related FixPacks z/OS connector V8.5.0, V8.5.1 or V8.6.0 and related FixPacks	“Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector.”
Fault-tolerant agent V8.5.0, V8.5.1 or V8.6.0 and related FixPacks z/OS connector V8.5.0, V8.5.1 or V8.6.0 and related FixPacks	“Upgrading the fault-tolerant agent and the z/OS connector” on page 216
dynamic domain manager V8.6.0 and related FixPacks Dynamic Workload Console V8.6.0 and related FixPacks z/OS connector V8.6.0 and related FixPacks	“Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector” on page 217.
dynamic domain manager V8.6.0 and related FixPacks z/OS connector V8.6.0 and related FixPacks	“Upgrading the dynamic domain manager and the z/OS connector” on page 218.
Dynamic Workload Console V8.5.0, V8.5.1 or V8.6.0 and related FixPacks z/OS connector V8.5.0, V8.5.1 or V8.6.0 and related FixPacks	“Procedure to upgrade the Dynamic Workload Console and the z/OS connector” on page 219.

Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector

How to upgrade the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector installed in the same directory.

About this task

If you installed a multiple components instance that contains a Dynamic Workload Console, a z/OS connector, and a fault-tolerant agent in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

The Dynamic Workload Console upgrade process migrates also the z/OS connector configuration properties.

If you try to upgrade the fault-tolerant agent first, the **twsinst** script stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
4. Upgrade the fault-tolerant agent by using the **twsinst** script as described in Upgrading agents and domain managers in *IBM Workload Scheduler: Planning and Installation*.

Upgrading the fault-tolerant agent and the z/OS connector

About this task

If you have a multiple components instance that contains a z/OS connector and a fault-tolerant agent in the directory <TWS_INST_DIR>, you must first configure the old z/OS connector in the <DWC_NEW_INST_DIR> directory in which you installed a new instance of the Dynamic Workload Console, uninstall the old z/OS connector and then you can upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

If you try to upgrade the fault-tolerant agent first, the **twsinst** script stops at the beginning and issues an error message that tells you the correct order on which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Install a Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in Chapter 13, “Installing,” on page 259.
2. Export the z/OS connector configuration properties in the old multiple components instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\displayZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/displayZosEngine.sh
```

For more information about the **displayZosEngine** tool, see *IBM Workload Scheduler for z/OS: Planning and Installation*.

Note: If you are connected to multiple controllers, repeat this step for each connection that you want to maintain in the new z/OS connector configuration.

3. Import the z/OS connector configuration properties into the Dynamic Workload Console instance, by running:

On Windows operating systems

<TWS_INST_DIR>\wastools\createZosEngine.bat

On UNIX and Linux operating systems

<TWS_INST_DIR>/wastools/createZosEngine.sh

For more information about the **createZosEngine** tool, see *IBM Workload Scheduler for z/OS: Planning and Installation*.

Note: If you want to maintain the connections to multiple controllers, repeat this step for each connection previously defined that you want to save.

4. Manually uninstall the old z/OS connector in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the earlier versions.
5. Upgrade the fault-tolerant agent by using the **twinst** script as described in *Upgrading agents and domain managers*.

Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector

How to upgrade the dynamic domain manager, the Dynamic Workload Console and the z/OS connector installed in the same directory.

About this task

If you installed a multiple component instance that contains a Dynamic Workload Console, a z/OS connector, and a dynamic domain manager in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory and then upgrade the dynamic domain manager in the <TWS_INST_DIR> directory.

The Dynamic Workload Console upgrade process migrates also the z/OS connector configuration properties.

If you try to upgrade the dynamic domain manager first, the installation process stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
4. Upgrade the dynamic domain manager, as described in *Upgrading a dynamic domain manager instance or its backup in IBM Workload Scheduler: Planning and Installation*.

Upgrading the dynamic domain manager and the z/OS connector

About this task

If you have a multiple components instance that contains a z/OS connector and a dynamic domain manager installed in the directory <TWS_INST_DIR>, you must first configure the old z/OS connector in the <DWC_NEW_INST_DIR> directory where you installed a new instance of the Dynamic Workload Console, uninstall the old z/OS connector, and then you upgrade the dynamic domain manager in the <TWS_INST_DIR> directory.

If you try to upgrade the dynamic domain manager first, the installation process stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Install a Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in Chapter 13, “Installing,” on page 259.
2. Export the z/OS connector configuration properties in the old shared instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\displayZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/displayZosEngine.sh
```

For more information about the **displayZosEngine** tool, see *IBM Workload Scheduler for z/OS: Planning and Installation*.

Note: If you are connected to multiple controllers, you have to repeat this step for each connection you want to maintain in the new z/OS connector configuration.

3. Import the z/OS connector configuration properties in the Dynamic Workload Console instance, by running:

On Windows operating systems

```
<TWS_INST_DIR>\wastools\createZosEngine.bat
```

On UNIX and Linux operating systems

```
<TWS_INST_DIR>/wastools/createZosEngine.sh
```

For more information about the **createZosEngine** tool, see *IBM Workload Scheduler for z/OS: Planning and Installation*.

Note: If you want to maintain the connections to multiple controllers, repeat this step for each connection previously defined that you want to save.

4. Manually uninstall the old z/OS connector in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the earlier versions.
5. Upgrade the dynamic domain manager as described in Upgrading a dynamic domain manager instance or its backup .

Procedure to upgrade the Dynamic Workload Console and the z/OS connector

About this task

If you installed a multiple components instance that contains a Dynamic Workload Console and a z/OS connector in the directory <TWS_INST_DIR>, you must upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory.

The Dynamic Workload Console upgrade process also migrates the z/OS connector configuration properties.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the old versions.
3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the old versions.

Managing z/OS engines by using WebSphere Application Server tools

You can manage z/OS engines by using WebSphere Application Server tools.

About this task

You can manage z/OS engines by using the WebSphere Application Server tools that are located in the following directory:

On Windows operating systems:

<DWC_INST_DIR>\wastools

On UNIX and Linux operating systems:

<DWC_INST_DIR>/wastools

Where <DWC_INST_DIR> is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.

- To create a z/OS connector instance by using the **createZosEngine** tool, see “Creating a z/OS connector instance” on page 220.
- To display the properties of a z/OS connector instance by using the **displayZosEngine** tool, see “Displaying the properties of a z/OS connector instance” on page 221.
- To update the properties of a z/OS connector instance by using the **updateZosEngine** tool, see “Updating the properties of a z/OS connector instance” on page 222.
- To list all the z/OS connector instances by using the **listZosEngine** tool, see “Listing all the z/OS connector instances” on page 224.
- To remove a z/OS connector instance by using the **removeZosEngine** tool, see “Removing a z/OS connector instance” on page 224.

Creating a z/OS connector instance

You can create a z/OS connector instance, if you want to connect to an IBM Workload Scheduler for z/OS by using a Dynamic Workload Console.

About this task

To create a z/OS connector instance, perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the z/OS connector instance with a Dynamic Workload Console.
2. Go to the `<DWC_INST_DIR>\wastools` directory, where `<DWC_INST_DIR>` is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.
3. Stop the application server by running the following script:

On Windows operating systems:

```
stopWas.bat
```

On UNIX and Linux operating systems:

```
stopWas.sh -user <TWSUser>  
-password <TWSUser_password>
```

where `<TWSUser>` is the IBM Workload Scheduler user for which you installed the z/OS connector instance and `<TWSUser_password>` is the IBM Workload Scheduler user password.

4. Create a file that contains the current security properties and from the `<DWC_INST_DIR>\wastools` run the following script:

On Windows operating systems:

```
createZosEngine.bat -name engineName  
-hostName hostName -portNumber portNumber  
[-connectionTimeoutCleanup cleanup_timeout]  
[-maxConnections maxconnections_number]  
[-connectionTimeout connection_timeout]  
[-unusedTimeout unused_timeout] [-reapTime reap_timeout]  
[-scaffoldSwitch scaffoldSwitch_boolean] [-useSsl ssl_boolean]
```

On UNIX and Linux operating systems:

```
createZosEngine.sh -name engineName  
-hostName hostName -portNumber portNumber  
[-connectionTimeoutCleanup cleanup_timeout]  
[-maxConnections maxconnections_number]  
[-connectionTimeout connection_timeout]  
[-unusedTimeout unused_timeout] [-reapTime reap_timeout]  
[-scaffoldSwitch scaffoldSwitch_boolean] [-useSsl ssl_boolean]
```

where

engineName

Specify the name of the IBM Workload Scheduler for z/OS engine that you are connecting to. Specify the name of the new z/OS connector instance.

hostName

Specify the host name or TCP/IP address of the remote z/OS system where the IBM Workload Scheduler for z/OS controller is installed.

portNumber

Specify the number of the TCP/IP port of the z/OS system that is used to communicate with the IBM Workload Scheduler for z/OS controller.

cleanup_timeout

Specify the connection timeout cleanup for the z/OS connection. This property is optional. The default value is 60 minutes.

maxconnections_number

Specify the maximum number of managed connections that can be created for the z/OS connection. This property is optional. The default value is 10.

connection_timeout

Specify the interval, in seconds, after which the z/OS connection request times out and a `connectionWaitTimeoutException` is thrown. This property is optional. The default value is 1 second.

unused_timeout

Specify the interval, in seconds, after which an unused connection is discarded by the connection pool maintenance thread. This property is optional. The default value is 60 seconds.

reap_timeout

Specify the reap time for the z/OS connection. This property is optional. The default value is 300 seconds.

scaffoldSwitch_boolean

Specify `true` to enable the scaffold switch for the z/OS connection or `false` to disable the property. This property is optional. The default value is `false`.

ssl_boolean

Specify `true` to enable the SSL communication between the z/OS connector and the remote z/OS system or `false` to disable the property. This property is optional. The default is `false`.

Note: If you set this value to `true`, ensure that the SSL is enabled for the z/OS controller. .

For more information about how to enable the SSL in the z/OS controller, see *IBM Workload Scheduler for z/OS Planning and Installation*:

Installing IBM Workload Scheduler for z/OS > Installing > Step 8. Securing communications

5. Start the application server by running the following script:

On Windows operating systems:

```
startWas.bat
```

On UNIX and Linux operating systems:

```
startWas.sh
```

Displaying the properties of a z/OS connector instance

You can display the properties of a z/OS connector instance if you want to see the contents.

About this task

To display the properties of a z/OS connector instance, perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the z/OS connector instance with a Dynamic Workload Console.

2. Go to the `<DWC_INST_DIR>\wastools` directory, where `<DWC_INST_DIR>` is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.
3. From the `<DWC_INST_DIR>\wastools` directory, run the following script:

On Windows operating systems:

```
displayZosEngine.bat engineName
```

On UNIX and Linux operating systems:

```
displayZosEngine.sh engineName
```

where `engineName` is the name of the IBM Workload Scheduler for z/OS engine that you are connecting to. Specify the name of the new z/OS connector instance.

Example

You receive the following output when you run the `displayZosEngine` tool for the `zos136` engine:

```
name : zos136
hostName : 9.87.130.95
portNumber : 20023
connectionTimeoutCleanup : 60 minutes
scaffoldSwitch : false
maxConnections : 10
connectionTimeout : 180 seconds
unusedTimeout : 1800 seconds
reapTime : 180 seconds
```

Updating the properties of a z/OS connector instance

You can update the properties of a z/OS connector instance if you want to change some properties values.

About this task

To update a z/OS connector instance, perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the z/OS connector instance with a Dynamic Workload Console.
2. Go to the `<DWC_INST_DIR>\wastools` directory, where `<DWC_INST_DIR>` is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.
3. Stop the application server by running the following script:

On Windows operating systems:

```
stopWas.bat
```

On UNIX and Linux operating systems:

```
stopWas.sh -user <TWSUser>
           -password <TWSUser_password>
```

where `<TWSUser>` is the IBM Workload Scheduler user for which you installed the z/OS connector instance and the `<TWSUser_password>` is the IBM Workload Scheduler user password.

4. From the `<DWC_INST_DIR>\wastools` run the following script:

On Windows operating systems:

```
updateZosEngine.bat -name engineName
-hostName hostName -portNumber portNumber
[-connectionTimeoutCleanup cleanup_timeout]
[-maxConnections maxconnections_number]
[-connectionTimeout connection_timeout]
[-unusedTimeout unused_timeout] [-reapTime reap_timeout]
[-scaffoldSwitch scaffoldSwitch_boolean] [-useSsl ssl_boolean]
```

On UNIX and Linux operating systems:

```
updateZosEngine.sh -name engineName
-hostName hostName -portNumber portNumber
[-connectionTimeoutCleanup cleanup_timeout]
[-maxConnections maxconnections_number]
[-connectionTimeout connection_timeout]
[-unusedTimeout unused_timeout] [-reapTime reap_timeout]
[-scaffoldSwitch scaffoldSwitch_boolean] [-useSsl ssl_boolean]
```

where

engineName

Specify the name of the IBM Workload Scheduler for z/OS instance you are updating.

hostName

Specify the host name or TCP/IP address of the remote z/OS system where the IBM Workload Scheduler for z/OS controller is installed.

portNumber

Specify the number of the TCP/IP port of the z/OS system that is used to communicate with the IBM Workload Scheduler for z/OS controller.

cleanup_timeout

Specify the connection timeout cleanup for the z/OS connection. This property is optional. The default value is 60 minutes.

maxconnections_number

Specify the maximum number of managed connections that can be created for the z/OS connection. This property is optional. The default value is 10.

connection_timeout

Specify the interval, in seconds, after which the z/OS connection request times out and a `connectionWaitTimeoutException` is thrown. This property is optional. The default value is 1 second.

unused_timeout

Specify the interval, in seconds, after which an unused connection is discarded by the connection pool maintenance thread. This property is optional. The default value is 60 seconds.

reap_timeout

Specify the reap time for the z/OS connection. This property is optional. The default value is 300 seconds.

scaffoldSwitch_boolean

Specify true to enable the scaffold switch for z/OS connection or false to disable the property. This property is optional. The default value is false.

ssl_boolean

Specify true to enable the SSL communication between the z/OS connector and the remote z/OS system or false to disable the property. This property is optional. The default is false.

Note: If you set this value to true, ensure that the SSL is enabled for the z/OS controller. .

For more information about how to enable the SSL in the z/OS controller, see *IBM Workload Scheduler for z/OS Planning and Installation: Installing IBM Workload Scheduler for z/OS > Installing > Step 8. Securing communications*

5. Start the application server by running the following script:

On Windows operating systems:

```
startWas.bat
```

On UNIX and Linux operating systems:

```
startWas.sh
```

Listing all the z/OS connector instances

You can list all the z/OS connector instances if you want to check the instances number.

About this task

To list all the z/OS connector instances, perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed z/OS connector instance with a Dynamic Workload Console.
2. Go to the `<DWC_INST_DIR>\wastools` directory, where `<DWC_INST_DIR>` is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.
3. From the `<DWC_INST_DIR>\wastools` directory run the following script:

On Windows operating systems:

```
listZosEngine.bat
```

On UNIX and Linux operating systems:

```
listZosEngine.sh
```

Removing a z/OS connector instance

You can remove a z/OS connector instance if you do not need anymore.

About this task

To remove a z/OS connector instance, perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the z/OS connector instance with a Dynamic Workload Console.
2. Go to the `<DWC_INST_DIR>\wastools` directory, where `<DWC_INST_DIR>` is the directory on which the z/OS connector instance is installed with a Dynamic Workload Console.
3. Stop the application server by running the following script:

On Windows operating systems:

```
stopWas.bat
```

On UNIX and Linux operating systems:

```
stopWas.sh -user <TWSUser>  
-password <TWSUser_password>
```

where *<TWSUser>* is the IBM Workload Scheduler user for which you installed the z/OS connector instance and the *<TWSUser_password>* is the IBM Workload Scheduler user password.

4. From the *<DWC_INST_DIR>\wastools* directory run the following script:

On Windows operating systems:

```
removeZosEngine.bat engineName
```

On UNIX and Linux operating systems:

```
removeZosEngine.sh engineName
```

where *engineName* is the name of the IBM Workload Scheduler for z/OS engine that you are connecting to. Specify the name of the z/OS connector instance that you want to remove.

5. Start the application server by running the following script:

On Windows operating systems:

```
startWas.bat
```

On UNIX and Linux operating systems:

```
startWas.sh
```

Configuring the Dynamic Workload Console to work with a remote z/OS connector

You can configure the Dynamic Workload Console to work with a remote z/OS connector.

About this task

You have the following environment:

DWC1 Dynamic Workload Console installed on the WKS1 workstation:

- The *DWC1* Dynamic Workload Console is installed with the *zConn1* z/OS connector in the *<DWC1_INST_DIR>* directory.
- The Jazz for Service Management extension for WebSphere is installed in the *<JazzSM_INST_DIR_WKS1>* directory.
- The WebSphere Application Server is installed in the *<WAS_INST_DIR_WKS1>* directory.

DWC2 Dynamic Workload Console installed on the WKS2 workstation:

- The *DWC2* Dynamic Workload Console is installed with the *zConn2* z/OS connector in the *<DWC2_INST_DIR>* directory.
- The Jazz for Service Management extension for WebSphere is installed in the *<JazzSM_INST_DIR_WKS2>* directory.
- The WebSphere Application Server is installed in the *<WAS_INST_DIR_WKS2>* directory.

By default the *DWC1* Dynamic Workload Console is configured to work with the local *zConn1* z/OS connector and the *DWC2* Dynamic Workload Console is configured to work with the local *zConn2* z/OS connector.

You can configure the *DWC1* Dynamic Workload Console to work with the remote *zConn2* z/OS connector installed with the *DWC2* Dynamic Workload Console, by importing the *DWC1* Dynamic Workload Console certificates into the *DWC2*

Dynamic Workload Console truststore and by importing the *DWC2* Dynamic Workload Console certificates into the *DWC1* Dynamic Workload Console truststore.

Run the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the *DWC1* Dynamic Workload Console.
2. Locate the SSL *DWC1* Dynamic Workload Console certificates, by performing the following steps:

- a. Run the following script:

On Windows operating systems:

```
showSecurityProperties.bat > My_Security1.prop
```

On UNIX and Linux operating systems:

```
showSecurityProperties.sh > My_Security1.prop
```

- b. In the *My_Security1.prop* file SSL Panel section , take note of the *keyFileName* and *trustFileName* values for the truststore and keystore certificates:

If you use the default certificates:

```
#####  
SSL Panel  
#####  
alias=NodeDefaultSSLSettings  
keyFileName=  
{CONFIG_ROOT}/cells/JazzSMNode01Cell/nodes/JazzSMNode01/key.p12  
keyFilePassword=*****  
keyFileFormat=PKCS12  
trustFileName=  
{CONFIG_ROOT}/cells/JazzSMNode01Cell/nodes/JazzSMNode01/trust.p12  
trustFilePassword=*****  
trustFileFormat=PKCS12  
clientAuthentication=false  
securityLevel=HIGH  
enableCryptoHardwareSupport=false
```

If you use your own certificates saved in the <certs_dir_1>:

```
#####  
SSL Panel  
#####  
alias=NodeDefaultSSLSettings  
keyFileName=  
<certs_dir>/mykey.p12  
keyFilePassword=*****  
keyFileFormat=PKCS12  
trustFileName=  
<certs_dir_1>/mytrust.p12  
trustFilePassword=*****  
trustFileFormat=PKCS12  
clientAuthentication=false  
securityLevel=HIGH  
enableCryptoHardwareSupport=false
```

3. On the *DWC1* Dynamic Workload Console extract the <*DWC1_cert*>.arm certificate by performing the following steps:
- a. Run the <*WAS_INST_DIR_WKS1*>\Appserver\java_1.7_64\jre\bin\ikeyman command or use the *ikeyman* command provided by a Java instance on your machine.
 - b. On the *ikeyman* wizard panel, click **Open a key database file**.

- c. Select as Key database type the PKCS12 value, click **Browse**, and use the following values:

If you use the default certificates:

Location

insert the <JazzSM_INST_DIR_WKS1>\profile\config\cells\
JazzSMNode01Cell\nodes\JazzSMNode01\ directory name.

Filename

Insert the key.p12 file name.

If you use your own certificates saved in the <certs_dir_1> directory:

Location

Insert the <certs_dir_1> directory name.

Filename

Insert the value you specified in the **keyFileName** field in
step 2b on page 226, in the example mykey.p12.

- d. Click **OK**.
- e. Insert the password value in the Password prompt panel:

If you use the default certificates:

WebAS.

If you use your own certificates saved in the <certs_dir_1> directory:

Insert the password that you specified in keyFilePassword field of
step 2b on page 226.

- f. Click **OK**.
- g. In the drop-down list, select **Personal certificates**.
- h. Select the following certificate and click **Extract Certificate**:

If you use the default certificates:

The **default** certificate.

If you use your own certificates saved in the <certs_dir_1> directory:

Your own certificate.

- i. In the New panel, select as Data type the Base64-encoded ASCII data value and select as <DWC1_certificates_dir> the path where you want to save the <DWC1_cert>.arm certificate.
- j. Click **OK**.

4. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the *DWC2* Dynamic Workload Console.

5. Locate the SSL *DWC2* Dynamic Workload Console certificates, by performing the following steps:

- a. Run the following script:

On Windows operating systems:

```
showSecurityProperties.bat > My_Security2.prop
```

On UNIX and Linux operating systems:

```
showSecurityProperties.sh > My_Security2.prop
```

- b. In the My_Security2.prop file SSL Panel section , take note of the keyFileName and trustFileName values for the truststore and keystore certificates:

If you use the default certificates:

```
#####  
SSL Panel  
#####  
alias=NodeDefaultSSLSettings  
keyFileName=  
${CONFIG_ROOT}/cells/JazzSMNode01Cell/nodes/JazzSMNode01/key.p12  
keyFilePassword=*****  
keyFileFormat=PKCS12  
trustFileName=  
${CONFIG_ROOT}/cells/JazzSMNode01Cell/nodes/JazzSMNode01/trust.p12  
trustFilePassword=*****  
trustFileFormat=PKCS12  
clientAuthentication=false  
securityLevel=HIGH  
enableCryptoHardwareSupport=false
```

If you use your own certificates saved in the <certs_dir_2> directory:

```
#####  
SSL Panel  
#####  
alias=NodeDefaultSSLSettings  
keyFileName=  
<certs_dir_2>/mykey.p12  
keyFilePassword=*****  
keyFileFormat=PKCS12  
trustFileName=  
<certs_dir_2>/mytrust.p12  
trustFilePassword=*****  
trustFileFormat=PKCS12  
clientAuthentication=false  
securityLevel=HIGH  
enableCryptoHardwareSupport=false
```

6. In the *DWC2* Dynamic Workload Console truststore, import the <DWC1_cert>.arm file created in step 3i on page 227, by performing the following steps:
 - a. On the *WKS1* workstation, copy the <DWC1_certificates_dir>\<DWC1_cert>.arm certificate to the <DWC2_certificates_dir> directory of the *WKS2* workstation.
 - b. Run the <WAS_INST_DIR_WKS2>\Appserver\java_1.7_64\jre\bin\ikeyman command.
 - c. In the ikeyman wizard panel, click **Open a key database file**.
 - d. Select as Key database type the PKCS12 value, Click **Browse** and use the following values:

If you use the default certificates:

Location

Insert the <JazzSM_INST_DIR_WKS2>\profile\config\cells\
JazzSMNode01Cell\nodes\JazzSMNode01\ directory name.

Filename

Insert the trust.p12 file name.

If you use your own certificates saved in the <certs_dir_2> directory:

Location

Insert the <certs_dir_2> directory name.

Filename

Insert the value you specified in the **trustFileName** field in the step 5b on page 227, in the example mytrust.p12.

- e. Click **OK**.

- f. Insert the password value in the Password prompt panel:

If you use the default certificates:

WebAS.

If you use your own certificates saved in the <certs_dir_2> directory:

Insert the password that you specified in trustFilePassword field of step 5b on page 227.

- g. Click **OK**.
 - h. Select **Signed certificates** in the drop-down list and click **Add**.
 - i. In the panel that opens, click **Browse** and select the <DWC2_certificates_dir>\<DWC1_cert>.arm certificate.
 - j. Click **OK**.
7. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the *DWC2* Dynamic Workload Console.
 8. On the *DWC2* Dynamic Workload Console, extract the <DWC2_cert>.arm certificate by performing the following steps:
 - a. Run the <WAS_INST_DIR_WKS2>\Appserver\java_1.7_64\jre\bin\ikeyman or use the ikeyman provided by a Java instance on your machine.
 - b. On the ikeyman wizard panel, click **Open a key database file**.
 - c. Select as Key database type the PKCS12 value, click **Browse**, and use the following values:

If you use the default certificates:

Location

Insert the <JazzSM_INST_DIR_WKS2>\profile\config\cells\JazzSMNode01Cell\nodes\JazzSMNode01\ directory name.

Filename

Insert the key.p12 file name.

If you use your own certificates saved in the <certs_dir_2> directory:

Location

Insert the <certs_dir_2> directory name.

Filename

Insert the value you specified in the **keyFileName** field in step 5b on page 227, in the example mykey.p12.

- d. Click **OK**.
- e. Insert the password value in the Password prompt panel:

If you use the default certificates:

WebAS.

If you use your own certificates saved in the <certs_dir_2> directory:

Insert the password that you specified in keyFilePassword field of step 5b on page 227.

- f. Click **OK**.
- g. Select **Personal certificates** in the drop-down list.
- h. Select the following certificate and click **Extract Certificate**:

If you use the default certificates:

The **default** certificate.

If you use your own certificates saved in the <certs_dir_2> directory:

Your own certificate.

- i. In the New panel, select as Data type the Base64-encoded ASCII data value, select <DWC1_certificates_dir> the path where you want to save the <DWC2_cert>.arm certificate.
- j. Click **OK**.
9. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed the *DWC1* Dynamic Workload Console.
10. On the *DWC1* Dynamic Workload Console truststore, import the <DWC2_cert>.arm created in step 8i, by performing the following steps:
 - a. From the *WKS2* workstation, copy the <DWC2_certificates_dir>\<DWC2_cert>.arm certificate to the <DWC1_certificates_dir> directory of the *WKS1* workstation.
 - b. Run the <WAS_INST_DIR_WKS1>\Appserver\java_1.7_64\jre\bin\ikeyman command.
 - c. In the ikeyman wizard panel, click **Open a key database file**.
 - d. Select as Key database type the PKCS12 value, click **Browse** and use the following values:

If you use the default certificates:

Location

Insert the <JazzSM_INST_DIR_WKS1>\profile\config\cells\
JazzSMNode01Cell\nodes\JazzSMNode01\ directory name.

Filename

Insert the trust.p12 file name.

If you use your own certificates saved in the <certs_dir_1> directory:

Location

Insert <certs_dir_1> directory name.

Filename

Insert the value that you specified in the **trustFileName** field in the step 2b on page 226, in the example mytrust.p12.

- e. Click **OK**.
- f. Insert the password value in the Password prompt panel:

If you use the default certificates:
WebAS.

If you use your own certificates saved in the <certs_dir_1>:
Insert the password that you specified in trustFilePassword field of step 2b on page 226.
- g. Click **OK**.
- h. Select **Signed certificates** in the drop-down list and click **Add**.
- i. In the panel that opens, click **Browse** and select the <DWC1_certificates_dir>\<DWC2_cert>.arm certificate.
- j. Click **OK**.
11. Verify that the *VMMRealm* value in the My_Security1.prop is different from the *VMMRealm* value in the My_Security2.prop file:

```
#####
Federated Repository Panel
#####
VMMRealm=defaultWIMFileBasedRealm1
#####
Federated Repository Panel
#####
VMMRealm=defaultWIMFileBasedRealm2
```

Note: If the *VMMRealm* value in *My_Security2.prop* is the same as the *VMMRealm* value in the *My_Security1.prop*, change one of them by running the **ChangeSecurityProperties.bat** on Windows operating systems or **ChangeSecurityProperties.sh** on UNIX and Linux operating systems. For more information about this utility, see *Administration Guide > Administrative tasks > Application server tasks*.

12. Stop the *DWC1* Dynamic Workload ConsoleWebSphere Application Server. For more information about this utility, see *Administration Guide > Administrative tasks > Application server tasks*.
13. Start the *DWC1* Dynamic Workload ConsoleWebSphere Application Server. For more information about this utility, see *Administration Guide > Administrative tasks > Application server tasks*.
14. Stop the *DWC2* Dynamic Workload ConsoleWebSphere Application Server. For more information about this utility, see *Administration Guide > Administrative tasks > Application server tasks*.
15. Start the *DWC2* Dynamic Workload ConsoleWebSphere Application Server. For more information about this utility, see *Administration Guide > Administrative tasks > Application server tasks*.

Encrypting the connection between the z/OS connector and the server started task

An overview of encrypting the connection between the z/OS connector and the server started task.

SSL encryption can be enabled to protect communication between the z/OS connector and the IBM Workload Scheduler for z/OS server started task.

To enable the use of the SSL default certificates provided with IBM Workload Scheduler for z/OS or your own certificates, you must:

1. On the UNIX System Services of the z/OS system where the server runs, use the *gskkyman* utility of z/OS Cryptographic Services System SSL to create the keystore database and generate the password file. Following this, you can import the default SSL certificates from the z/OS connector or from the Dynamic Workload Console.
2. Configure IBM Workload Scheduler for z/OS by specifying the *TCPOPTS* statement for the server started task.

See “Security for TCP/IP connections” on page 99 for details.

Chapter 8. Installing and uninstalling on WebSphere Application Server for z/OS

How to install and uninstall on WebSphere Application Server for z/OS.

This chapter describes how to install, apply maintenance, and uninstall the IBM Workload Scheduler for z/OS connector on IBM WebSphere Application Server for z/OS. Install the z/OS connector to maintain your business in a z/OS environment and simultaneously manage your workload using modern applications like EJB and Web Services as described in “Business Scenario”, or to work with the Dynamic Workload Console as described in “Graphical user interfaces” on page 7.

Business Scenario

To save money, skill, and seize new opportunities, a company wants to maintain its business in a z/OS environment and simultaneously manage its workload using modern applications like Web Services. It wants to avoid using scripting languages to schedule jobs on non-IBM software applications, because these scripts can be hard to debug, maintain, and port across different operating systems.

The company can reach this objective by:

- Installing the IBM Workload Scheduler for z/OS connector on WebSphere Application Server for z/OS to have all the required functions available in the z/OS environment.
- Using the Java API to define and submit a jobs.

Authorization roles required for installing and uninstalling

To install or uninstall IBM Workload Scheduler for z/OS connector, you must have full access (read, write, and execute) to:

- The directory where you installed the WebSphere Application Server for z/OS. The default value is *WebSphere Application Server_installation_directory/AppServer/profiles/default/bin* directory. This is the directory where the **wsadmin.sh** script is located.
- The directories where you extract the JWSZ934 FMID.

Installing on WebSphere Application Server for z/OS

You can install the IBM Workload Scheduler for z/OS connector on WebSphere Application Server for z/OS either by using the Integrated Solution Console or the **zConnInstall.sh** scripts located in theopt\package directory.

Integrated Solution Console

It is a graphical interface to manage your applications and perform system administration tasks for your WebSphere Application Server environment. The administrative console runs in your web browser. You can use it to manage WebSphere Application Server applications. Refer to “Installing using the Integrated Solution Console” on page 234.

The zConnInstall.sh script

It uses the WebSphere Application Server **wsadmin** tool to perform the installation. Refer to “Installing using the zConnInstall.sh script” on page 235.

The IBM Workload Scheduler for z/OS connector is installed using the same user with which you access the WebSphere Application Server for z/OS.

Installing using the Integrated Solution Console

To install IBM Workload Scheduler for z/OS connector with the Integrated Solution Console, perform the following steps:

1. Ensure that you installed the JWSZ934 FMID.
2. Install the resource adapter file.
 - a. Select **Resource->Resource Adapters->Resource adapters**.
 - b. On the Resource adapters dialog, select **Install RAR** both to install a RAR file and to configure an associated resource adapter.
 - c. On the Install RAR dialog, in the **Node** field, set the scope to a WebSphere Application Server node on which the resource adapter must be installed.
 - d. Browse to find the appropriate RAR file. If your RAR file is located on your:
 - Local workstation, select **Local file system**, and browse to find the file.
 - Server, select **Remote file system**, and specify the fully qualified path to the file.

Repeat Step 2 for each node that hosts application servers in the cluster.

3. Create the connection factory for an IBM Workload Scheduler engine.
 - a. Select **Resource->Resource Adapters->J2C connection factories**.
 - b. On the J2C connection factories page, in the **Scope** pull-down menu, set the scope to the WebSphere Application Server node where you installed the resource adapter.
 - c. Select **New**.
 - d. On the configuration page, specify the following settings:

Table 38. Configuration page settings

Setting	Value
Provider	ZOSConnectorAdapter. It is the name of the resource adapter that you specified when you installed the resource adapter.
Name	The name of the engine that you use to connect to the Dynamic Workload Console.

- e. Click **Apply**.
- f. On the Custom properties page, in the Additional Properties list, click **Custom properties**.
- g. From the table displayed, select **HostName**.
- h. On the General Properties page, in the **Value** field, enter the host name of the IBM Workload Scheduler for z/OS server and then click **OK**.
- i. On the **Custom properties** table, select **PortNumber**.
- j. On the General Properties page, in the **Value** field enter the port number of the IBM Workload Scheduler for z/OS server and then click **OK**.
- k. In the **JNDI name** field, replace file path `eis/zos_engine_name` with `eis/tws/zconn/zos_engine_name` and then click **OK**.

- l. In the Messages box, click **Save** to save your changes directly to the master configuration.
4. Install the connector enterprise application.
 - a. Select **Applications->Applications Types->WebSphere enterprise applications**.
 - b. On the Enterprise Applications page, click **Install**.
 - c. Specify the path to the ZConnector.ear file.
Browse to find the appropriate EAR file. If your EAR file is located on your:
 - Local workstation, select **Local file system**, and browse to find the file.
 - Server, select **Remote file system**, and specify the fully qualified path to the file.
 - d. Deploy the application on the cluster with the name **ZConnector**.
5. Copy the TWSZOSConnConfig.properties file in the `<installation_directory>/AppServer/profiles/default/properties` path.
6. Identify the library files and their classpath by performing the following steps:
 - a. In the console navigation tree, click **Environment->Shared Libraries**. The Shared libraries page is displayed.
 - b. Change the scope of the collection table to see the shared libraries the are in a cell, in a node, or in a server.
 - c. Select the cell, the node or the server where you find the shared library and click **New**.
 - d. On the Configuration page for a shared library, specify the name and the classpath for the library as follows:
Name=applicationJobPlugins
ClassPath=/zConn_instdir/package/apps/applicationJobPlugins
 - e. Click **OK**.
7. Associate the applicationJobPlugins shared library to the ZConnector application by performing the following steps:
 - a. Click **Applications->Applications Types->WebSphere enterprise applications** in the console navigation tree. The General Properties page is displayed.
 - b. Click the installed application ZConnector.
 - c. Click **Shared libraries references**. The Shared libraries references page is displayed.
 - d. From the Shared libraries references list select ZConnector. The Available list is displayed.
 - e. In the Available list, select the applicationJobPlugins library.
 - f. Click **OK**.

The installation does not show the path where you installed the z/OS connector. The path is specified in the **com.ibm.ws.scripting.traceFile** property in the wsadmin.properties file. The connector enterprise application is installed in the *WebSphere Application Server_installation_directory*/AppServer/profiles/default/InstalledApps directory.

Installing using the zConnInstall.sh script

To install IBM Workload Scheduler for z/OS connector with the zConnInstall.sh script, perform the following steps:

1. Ensure that you installed the JWSZ934 FMID.

- Run the `zConnInstall.sh` script. The script requires the path where you installed the WebSphere Application Server for z/OS and the properties described in Table 39 to run.

Table 39. `zConnInstall.properties` properties and corresponding values

Property	Value
CellName	The cell name of the profile. This property is required.
nodeName	The node name of the profile. This property is required.
serverName	The server name of the profile. This property is required.
zosEngineName	The engine name for z/OS connection. This property is required.
zosHostName	The host name of the IBM Workload Scheduler for z/OS server. This property is required.
zosPortNumber	The port number of the IBM Workload Scheduler for z/OS server. This property is required.
connectionTimeoutCleanup	The connection timeout cleanup for z/OS connection. This property is optional. The default value is 10 minutes.
scaffoldSwitch	The scaffold switch for z/OS connection. This property is optional. The default value is false .
maxConnections	The maximum number of the managed physical connections that can be created for the z/OS in the connection pool. This property is optional. The default value is 40 . Note: If the Dynamic Workload Console Client makes a request, a connection between the IBM Workload Scheduler for z/OS connector and the z/OS Dynamic Workload Console server is created. Remember that if you click a different object in the Dynamic Workload Console, the Dynamic Workload Console Client creates another request and then a new connection. When the <code>maxConnections</code> value is reached, no new physical connection is created and the process that requests the connection, waits until a used connection is released to use it.
connectionTimeout	The interval, in seconds, after which a connection request goes in timeout and a <code>ConnectionWaitTimeoutException</code> error is detected in the z/OS. The timeout is necessary if the <code>maxConnections</code> value for the connection pool is reached. This property is optional. The default value is 180 seconds. Note: For example, if the <code>connectionTimeout</code> value is set to 30 seconds and the maximum number of connections is reached, the Pool Manager waits for 30 seconds for an available physical connection. If a physical connection is not available within this time, the Pool Manager throws a <code>ConnectionWaitTimeoutException</code> .
unusedTimeout	The interval, in seconds, after which an unused or idle connection is discarded. Set this value greater than the <code>reapTime</code> value. This property is optional. The default value is 300 seconds.

Table 39. *zConnInstall.properties* properties and corresponding values (continued)

Property	Value
reapTime	The interval, in seconds, between the runs of the pool maintenance thread. The pool maintenance thread runs every <i>reapTime</i> seconds and should be set to a value less than the value of <i>unusedTimeout</i> . The maintenance thread discards any connection that remains unused for longer than the <i>unusedTimeout</i> . This value can affect performance since smaller <i>reapTime</i> means that the maintenance thread runs more often and thereby using more resources. This property is optional. The default value is 60 seconds.
startApplication	Start the WebSphere Enterprise Applications. This property is optional. Possible values are true and false. The default value is false .

You can run the script in one of the following ways:

- Specifying the path where you installed the WebSphere Application Server for z/OS and the properties directly in the command line. In this case the path where you installed the WebSphere Application Server for z/OS must be specified as the first property. The following example shows how to run the command specifying all the parameters:

```
./zConnInstall.sh -wasPath /u/wasv7config/bbobase/bbonode/AppServer
-cellName bbobase -nodeName bbonode -serverName server1 -zosEngineName CWSV64
-zosHostName 127.0.0.1 -zosPortNumber 505 -connectionTimeoutCleanup 10
-scaffoldSwitch false -maxConnections 40 -connectionTimeout 180
-unusedTimeout 300 -reapTime 60 -startApplication false
```

- Specifying in the *zConnInstall.properties* file all the values of the properties and running the command specifying only the path where you installed the WebSphere Application Server for z/OS as follows:

```
./zConnInstall.sh -wasPath app_server_root
```

3. Verify that the installation completed successfully by reading the messages displayed in the screen. Detailed information is logged in the file specified in the **com.ibm.ws.scripting.traceFile** property in the *wsadmin.properties* file.

The installation does not show the path where you installed the z/OS connector. The path is specified in the **com.ibm.ws.scripting.traceFile** property in the *wsadmin.properties* file. By default, the connector enterprise application is installed in the *WebSphere Application Server_installation_directory/AppServer/profiles/default/InstalledApps* directory.

Note: The IBM Workload Scheduler for z/OS server has a limitation for the number of the physical connections and if the number of connection grows over, you can customize the IBM Workload Scheduler for z/OS connector to take down the number the connection with no Security issue for the Dynamic Workload Console users, see “Considerations” on page 156

Installation and uninstallation log files

You can check the following log files for information about the installation. Details of the installation process are displayed on the screen and kept in log files in the following directories:

If you installed using the Integrated Solution Console

The file path is specified by the **com.ibm.ws.scripting.traceFile** property in the *wsadmin.properties* file. The default value is **app_server_root/profiles/profiles_name/logs/server_name/was.traceout**.

where:

profiles_name

Is the name of the profile that you used when you installed the WebSphere Application Server. The default name is **default**.

server_name

Is the name of the server that you used when you installed the WebSphere Application Server. The default name is **default**.

If you installed using the zConnInstall.sh script

The file path is specified by the **com.ibm.ws.scripting.traceFile** property in the `wsadmin.properties` file. The default value is **app_server_root/profiles/profiles_name/logs/was.traceout**, where *profiles_name* is the name of the profile that you used when you installed the WebSphere Application Server. The default name is **default**.

Install the additional plug-in

Install an additional plug-in for z/OS connector on WebSphere Application Server for z/OS

To manually install an additional plug-in for IBM Workload Scheduler for z/OS connector on WebSphere Application Server for z/OS, perform the following steps:

1. Ensure that you installed the JWSZ934 FMID.
2. Locate the path where you installed the WebSphere Application Server for z/OS.
3. Copy the file `<plug-in_jar-file>.jar` to the `sharedLibraries` directory of the WebSphere Application Server for z/OS installation.

Enabling communications with Dynamic Workload Console

After you installed the Dynamic Workload Console and the z/OS connector you must enable the communication between them. The Dynamic Workload Console and the z/OS connector use RMI/IIOP over SSL to communicate. The SSL security paradigm implemented in the WebSphere Application Server requires two stores to be present on the clients and the server:

A keystore

It contains the private key.

A truststore

It contains the certificates of the trusted counterparts.

Figure 27 on page 239 shows the keys that must be extracted and distributed to enable SSL between the z/OS connector and the Dynamic Workload Console. Each arrow in the diagram includes the following activities performed using an appropriate key management tool on each keystore:

- Create a self-signed certificate or import a third party certificate.
- Extract a new key.
- Open the appropriate truststore.
- Use the new key to add a signed certificate to the truststore.

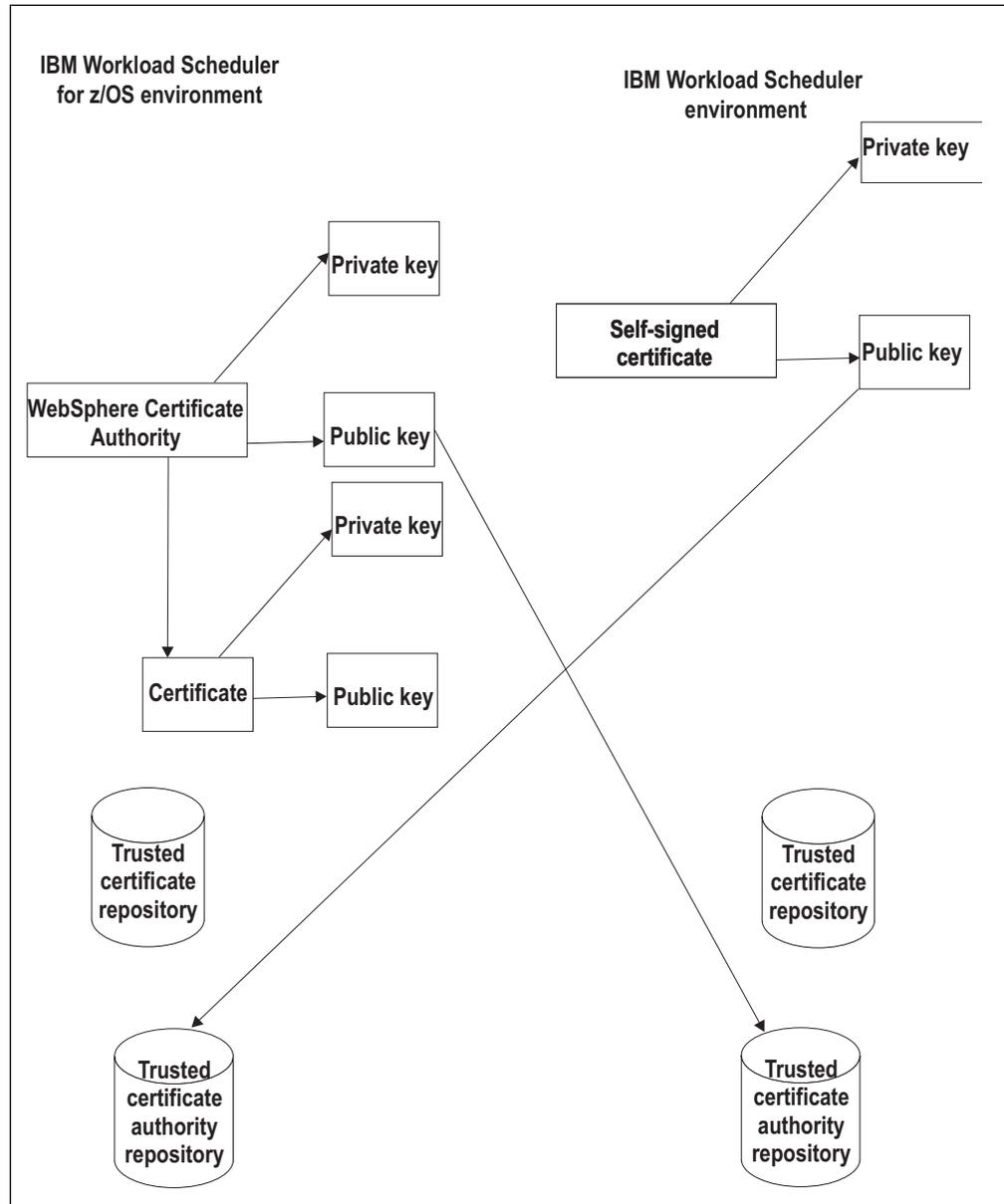


Figure 27. Shows the keys to enable SSL between the z/OS connector and the Dynamic Workload Console

To define SSL basic authentication security, you must first request a signed certificate for your server and a certificate authority (CA) certificate from the certificate authority that signed your server certificate. After you have received both these certificates, you must:

- From the z/OS environment, extract the public key CA certificate and store it in the trusted Certificate Authority repository of the Dynamic Workload Console.
- From the Dynamic Workload Console, extract the public key of the self-signed certificate and store it in the trusted certificate repository of WebSphere Application Server for z/OS.

To perform these operations, complete the following steps:

1. Export the WebSphere Application Server for z/OS certificate to a data set, as follows:

- a. Connect to RACF and select option DIGITAL CERTIFICATES, KEY RINGS, AND TOKENS.
 - b. Select option Digital certificates functions.
 - c. Select option Write a certificate to a data set.
 - d. Export the WebSphere Application Server certificate authority certificate to a data set and transfer the file to the Dynamic Workload Console using the FTP protocol in binary or ASCII mode.
2. Import the file into the trusted certificate authority repository of the Dynamic Workload Console using the iKeyman utility. The iKeyman utility is located in *installation_directory*/TDWC/_jvm/jre/bin.
 3. From the Dynamic Workload Console, export the self-signed certificate to a file using the iKeyman utility. For more information, see the section about interface communication in Administration Guide.
 4. Transfer the file to the z/OS environment and add it to the RACF database as follows:
 - a. In RACF, select option DIGITAL CERTIFICATES, KEY RINGS, AND TOKENS.
 - b. Select option Digital certificates functions.
 - c. Select option Add, alter, delete or list certificates.
 - d. Select option Add a digital certificate to the RACF database. Set the status to **Trust (T)**.
 5. Associate the certificate to the trusted certificate authority repository of WebSphere Application Server for z/OS, as follows:
 - a. In RACF, select option Key Ring functions.
 - b. Select option Connect a digital certificate to a key ring. In field Ring Name, type the name of the WebSphere Application Server controller key ring.
 6. Define an EJBROLE profile and then permit a System Authorization Facility (SAF) user to the profile as follows:
 - a. On the WebSphere Application Server, the deployment descriptor of the zConnector defined under the Enterprise Applications, displays the default role, TWSAdmin, that needs to be defined in the RACF class EBJROLE as follows:


```
rdefine EJBROLE <SAF_prefix>.TWSAdmin owner(SYS1)
audit(failures(READ)) uacc(NONE)
```

where, <SAF_prefix> is the prefix of the profile, and the value can be found on the WebSphere Application Server, **Security > Global security > External authorization providers > SAF authorization options**, in the SAF profile prefix field.
 - b. Grant READ access to a specific user by issuing the following RACF command:


```
permit <SAF_prefix>.TWSAdmin class(EJBROLE) id(userid) access(READ)
```
 7. Restart WebSphere Application Server to make changes effective.

Secure communications is now enabled between the Dynamic Workload Console and the z/OS connector.

Applying maintenance

This section describes how to apply the program temporary fix (PTF) level of the IBM Workload Scheduler for z/OS connector. You can apply PTFs using either the Integrated Solution Console, or the **zConnUpdate.sh** scripts located in the *install_dir/zConnUpdate.sh* directory. This section describes how to apply the PTFs using the **zConnUpdate.sh** scripts.

1. Apply the PTFs as described in the *Program Directory for IBM Workload Scheduler for z/OS*.
2. Ensure that the WebSphere Application Server for z/OS Integrated Solution Console is not running.
3. Ensure that you installed the JWSZ934 FMID.
4. Move to the directory where the script is located. The default value is *installation directory/opt/package*.
5. Ensure that the **com.ibm.ws.scripting.connectionType** property in the *wsadmin.properties* file is set to the value **SOAP** or **RMI**. The default value is **SOAP**. The script requires the WebSphere Application Server user and password. If you do not want to specify them at run time, set them in the *soap.client.props* file for SOAP connection type and in the *sas.client.props* file for the RMI connection type. The *soap.client.props* and *sas.client.props* files are located in the *properties* directory of your WebSphere Application Server profile. The *WebSphere Application Server_installation_directory/AppServer/profiles/WebSphere Application Serverprofile_name/properties* is the default directory.
6. Run the **zConnUpdate.sh** script. The script requires the path where you installed the WebSphere Application Server for z/OS and the properties described in Table 39 on page 236 to run.

Table 40. *zConnUpdate.properties* properties and corresponding values

Property	Value
CellName	The cell name of the profile. This property is required.
nodeName	The node name of the profile. This property is required.
serverName	The server name of the profile. This property is required.

You can run the script in one of the following ways:

- Specifying the path where you installed the WebSphere Application Server for z/OS and the properties directly in the command line. In this case the path where you installed the WebSphere Application Server for z/OS must be specified as the first property. The following example shows how to run the command specifying all the parameters:

```
./zConnUpdate.sh -wasPath /u/wasv7config/bbobase/bbonode/AppServer  
-cellName bbobase -nodeName bbonode -serverName server1 -zosEngineName CWSV64
```
 - Specifying in the *zConnUpdate.properties* file all the values of the properties and running the command specifying only the path where you installed the WebSphere Application Server for z/OS as follows:

```
./zConnUpdate.sh -wasPath app_server_root
```
7. Verify that the PTF was installed successfully by reading the messages displayed on the screen. Detailed information is logged in the file specified in the **com.ibm.ws.scripting.traceFile** property in the *wsadmin.properties* file.

Uninstalling

To uninstall the IBM Workload Scheduler for z/OS connector, you can use either the Integrated Solution Console or the `zConnUninstall.sh` script.

Uninstalling using the Integrated Solution Console

To uninstall the IBM Workload Scheduler for z/OS connector with the Integrated Solution Console, perform the following steps:

1. Uninstall the resource adapter file.
 - a. Select **Resource->Resource Adapters->Resource adapters**.
 - b. On the Resource adapters dialog, select the resource adapter that you want to delete and click **Delete**. A message is displayed. Click **Save**.

Repeat this step for each node that hosts application servers in the cluster.

2. If still present, uninstall the connection factory for an IBM Workload Scheduler engine.
 - a. Select **Resource->Resource Adapters->J2C connection factories**.
 - b. On the J2C connection factories page, in the **Scope** pull-down menu, set the scope to the WebSphere Application Server node where you installed the resource adapter.
 - c. Select the J2C connection factory that you want to delete and click **Delete**. A message is displayed. Click **Save**.
3. Delete the Connector enterprise application.
 - a. Select **Applications->Applications Types->WebSphere enterprise applications**.
 - b. On the Enterprise Applications page, select the ZConnector resource that you want to uninstall and click **Uninstall**. This step deletes the application from the product configuration repository and the application binaries from the file system of all nodes where the application modules are installed. A message is displayed. Click **Save**.

Uninstalling using the `zConnUninstall.sh` script

To uninstall IBM Workload Scheduler for z/OS connector with the `zConnUninstall.sh` script, perform the following steps:

1. Ensure that you installed the JWSZ934 FMID.
2. Ensure that the `com.ibm.ws.scripting.connectionType` property in the `wsadmin.properties` file is set to the **SOAP** or **RMI** value. The default value is **SOAP**. In this case at run time, the script requires the WebSphere Application Server user and password. If you do not want to specify them at run time, set them in the `soap.client.props` file for SOAP connection type and in the `sas.client.props` file for the RMI connection type. The `soap.client.props` and `sas.client.props` files are located in the `properties` directory of your WebSphere Application Server profile. The `WebSphere Application Server_installation_directory/AppServer/profiles/WebSphere Application Serverprofile_name/properties` is the default directory.
3. Run the `zConnUninstall.sh` script. The script requires the path where you installed the WebSphere Application Server for z/OS and the properties described in Table 39 on page 236 to run.

Table 41. *zConnUninstall.properties* properties and corresponding values

Property	Value
CellName	The cell name of the profile. This property is required.
nodeName	The profile node name. This property is required.
serverName	The profile server name. This property is required.
zosEngineName	The engine name for z/OS connection. This property is required.

You can run the script in one of the following ways:

- Specifying the path where you installed the WebSphere Application Server for z/OS and the properties directly in the command line.

In this case the path where you installed the WebSphere Application Server for z/OS must be specified as the first property. The following example shows how to run the command specifying all the parameters:

```
./zConnUninstall.sh -wasPath /u/wasv7config/bbobase/bbonode/AppServer
-cellName bbobase -nodeName bbonode -serverName server1 -zosEngineName CWSV64
```

- Specifying in the `zConnUninstall.properties` file all the values of the properties and running the command specifying only the path where you installed the WebSphere Application Server for z/OS as follows:

```
./zConnUninstall.sh -wasPath app_server_root
```

4. Verify that the uninstallation completed successfully by reading the messages displayed on the screen. Detailed information is logged in the file specified in the `com.ibm.ws.scripting.traceFile` property in the `wsadmin.properties` file.

Uninstalling additional plug-ins

Uninstall an additional plug-in for z/OS connector on WebSphere Application Server for z/OS

To manually uninstall the additional plug-in on IBM Workload Scheduler for z/OS connector, perform the following steps:

1. Ensure that you installed the JWSZ934 FMID.
2. Locate the path where you installed the WebSphere Application Server for z/OS.
3. Delete your plug-in file `<plug-in_jar-file>.jar` from the `sharedLibraries` directory of the WebSphere Application Server for z/OS installation directory.

Chapter 9. Troubleshooting and maintaining the installation

How to troubleshoot and maintain the installation of the IBM Workload Scheduler for z/OS connector.

This chapter describes how to troubleshoot and maintain the installation of the IBM Workload Scheduler for z/OS connector.

Maintaining the installation of the z/OS connector

This section explains some maintenance procedures you should perform in specific situations.

Updating the SOAP properties after changing the WebSphere Application Server user or its password

If you change the user ID or the password of the WebSphere Application Server administrator, you must also update the SOAP client properties.

To update the properties, run the following command:

- On UNIX and Linux: **updateWas.sh** command from the *TWA_home/wastools* directory.
- On Windows: **updateWas.bat** command from the *TWA_home\wastools* directory.

Updating the SOAP properties usage

To update the SOAP properties, use the syntax in “updateWas.sh (.bat).”

After using this command you must restart the application server.

updateWas.sh (.bat)

Format

```
updateWas.sh (.bat) -user new_WAS_admin_user -password pwd
```

Parameters

```
-user new_WAS_admin_user -password pwd
```

Supply the user and password of the new WebSphere Application Server administration user that you want to be configured as the credentials in the SOAP client properties.

Part 4. Dynamic Workload Console

How to install, upgrade, configure, uninstall and troubleshoot the Dynamic Workload Console.

Chapter 10. Overview of the Dynamic Workload Console

An overview of the Dynamic Workload Console.

The Dynamic Workload Console is a web-based user interface that is used with the following products:

- IBM Workload Scheduler
- IBM Workload Scheduler for z/OS
- IBM Workload Scheduler for Applications
- Dynamic Workload Broker

You can access IBM Workload Scheduler and dynamic workload broker environments from any location in your network using one of the supported browsers connected to the Dynamic Workload Console. The Dynamic Workload Console must be installed on a system that can reach either the IBM Workload Scheduler or the dynamic workload broker nodes using network connections.

Chapter 11. Preparing

An overview on how to install and use the Dynamic Workload Console.

To install and use the Dynamic Workload Console :

1. Check the installation prerequisites in the Detailed System Requirements at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045182> to verify that your system is compliant.
2. Choose the installation method that best suits your needs as described in "Selecting your installation method" on page 259.
3. Collect the information necessary to type in the required fields during the installation. See Chapter 13, "Installing," on page 259.
4. Install the Dynamic Workload Console by following the instructions provided in "Installing components using Installation Manager."
5. Log in to the Dynamic Workload Console as described in "Accessing the Dynamic Workload Console" on page 272.
6. In the navigation tree on the left, click one of the following:

IBM Workload Scheduler

To access the IBM Workload Scheduler available functions

dynamic workload broker

To access the dynamic workload broker available functions

7. To effectively manage the functions available in the Dynamic Workload Console, create *engine connections* to the IBM Workload Scheduler and dynamic workload broker environments that you want to manage. Without defining engine connections, you can use only a limited set of Dynamic Workload Console functions. For more information, see "Quick steps to define an IBM Workload Scheduler engine connection" on page 274 and "Quick steps to define an dynamic workload broker connection" on page 274.

Directories created outside of *TWA_home* at installation time

The following list shows the directories that are created outside of *TWA_home* when you install the Dynamic Workload Console and IBM Workload Scheduler for z/OS connector.

On Windows operating systems:

Dynamic Workload Console:

%WINDIR%\TWA

z/OS connector:

%WINDIR%\TWA

%WINDIR%\system32\TWSRegistry.dat (32 bits)

%WINDIR%\syswow64\TWSRegistry.dat (32 bits on 64 bits)

%WINDIR%\TWSRegistry.dat (64 bits on 64 bits)

On UNIX operating systems:

Dynamic Workload Console:

/etc/TWA

z/OS connector:

Accessing the installation media

Accessing the installation media

Access the installation media to download the installation files

Using DVDs

About this task

In this installation scenario, you have the DVDs that contain the installation files, and typically, you install the product on your computer.

Install Dynamic Workload Console from DVDs, performing the following steps:

1. You must choose the Dynamic Workload Console DVD.
2. Insert the product installation DVD into the drive.

For a complete list of the Installation media, see the IBM Workload Scheduler Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

For more information about the media structure, see “Installation media.”

Installation media

The Dynamic Workload Console is packaged into multiple DVDs, one for each of the supported operating systems. Each DVD contains:

Dynamic Workload Console DISK

Depending on the operating system, the installation DVD contains some or all the following directories:

FULL Contains the repository necessary to install the product by using Installation Manager.

iim Contains the files necessary to install the Installation Manager by using the launchpad or manually.

Launchpad

Contains the launch pad code

response_files

Contains the response files to install Dynamic Workload Console.

TDWC

Contains the files necessary to install the Dynamic Workload Console.

For a complete list of DVDs and supported operating systems, see the Dynamic Workload Console Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

Downloading eImages on your workstation

About this task

You can download eImages, by performing the following steps:

1. Ensure that your workstation has sufficient space to store both the files you must download from IBM Passport Advantage® and the extracted installation image.
2. From IBM Passport Advantage, download all the required parts for the product image listed into the Table 42 to a temporary directory.

Table 42. Dynamic Workload Console required eImages

To install IBM Workload Scheduler components:	eImages to download:
Dynamic Workload Console	<ul style="list-style-type: none">• Dynamic Workload Console eImage.• WebSphere Application Server eImage.• Jazz for Service Management extension for WebSphere eImage.

3. Extract the installation image from the downloaded file and verify that the installation image is complete.

For more information about eImages, see the Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

Creating a network repository

About this task

This scenario is valid only for master domain manager or dynamic domain manager and their backups, or Dynamic Workload Console that are installed by using the Installation Manager infrastructure.

Use the Installation Manager **Package Utility** to create the IBM Workload Scheduler DVDs or eImages in network repository format. For more information about the **Package Utility** wizard, see *Installation Manager > Installing > Managing packages with Packaging Utility* in the Installation Manager Information center <http://pic.dhe.ibm.com/infocenter/install/v1r6/index.jsp>.

After you use the **Packaging Utility** to create a repository from the ESD images, you can use the Installation Manager to define this location as a repository. You can save the repository on a UNC drive on Windows operating systems or on a web server to make the directories and files available over HTTP.

To create an IBM Workload Scheduler network repository, perform the following procedure:

1. Download the eImages as described in Downloading eImages on your workstation or use the DVDs as described in Using DVDs in *IBM Workload Scheduler: Planning and Installation*.
2. Install Installation Manager on your workstation.
3. Install the **Package Utility** using Installation Manager on your workstation.
4. To create the ESD images in network format, run the following steps:
 - a. Start the **Package Utility**.

- b. Click **Point to the ESD image**.
- c. Run the wizard. For more information about the **Package Utility** wizard, see *Installation Manager > Installing > Managing packages with Packaging Utility*.

After you created a repository in network format, define this location as an Installation Manager repository. To add a repository, run the following steps:

1. Open the Installation Manager wizard.
2. Select **File > Preferences**. The Repositories page is displayed and shows available repositories, repository locations, and the connection status for the repositories.
3. Select **Add Repository**. The Add Repository page is displayed.
4. Enter the repository location or select **Browse**.
5. Go to the repository location where you saved the eImages or the DVD content in network format and select the URL related to the product that you want to install.
6. Click **OK**. If you provided an HTTPS or restricted FTP repository location, you are prompted to enter a user ID and password. The new repository location is added to the list. If the repository is not connected, a red box is shown in the Connection column.
7. Click **OK**.

After you defined an Installation Manager repository, install the product:

IBM Workload Scheduler

See Installing main components in *IBM Workload Scheduler: Planning and Installation*.

Dynamic Workload Console

See Chapter 13, "Installing," on page 259.

Chapter 12. Dynamic Workload Console prerequisites

Prerequisite information for installing a Dynamic Workload Console

Dynamic Workload Console installation has the following prerequisites:

WebSphere Application Server

If you do not have this product installed, the installation process automatically installs it.

Jazz for Service Management extension for WebSphere

If you do not have this product installed, the installation process automatically installs it.

Dashboard Application Services Hub

If you do not have this product installed, the installation process automatically installs it.

WebSphere SDK Java Technology Edition

If you do not have this product installed, the installation process automatically installs it.

To install the prerequisites, choose one of the following options:

- Use the launchpad. See "Launchpad".
- Manually launch the Jazz for Service Management extension for WebSphere installation on the product DVD.
- Download the appropriate eImages. See the product Download Document.

For a complete list of the correct versions to install, see the <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

Supported operating systems

What actions to perform to produce a dynamic report that lists the supported operating systems.

About this task

To produce a dynamic report that lists the supported operating systems, perform the following actions:

1. Go to the web page: <http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html>.
2. Click the *"Operating systems for a specific product"* report.
3. In the window "Operating systems for a specific product", type in the field *Enter a full or partial product name* with the value **IBM Workload Scheduler** and click *Search*.
4. In the *Select version* drop-down list, select version **9.3.0** and click *Submit to run the report*.

For a complete list of system requirements (disk spaces, temporary spaces and RAM usage), see System Requirements Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

Scanning system prerequisites for IBM Workload Scheduler

Before installing or upgrading the product, IBM Workload Scheduler automatically runs a scan on your system.

About this task

Having an environment that meets the product system requirements ensures that an installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- On UNIX operating systems, the necessary product libraries are installed.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory.

Note: The scan verifies only that the environment meets the requirements of IBM Workload Scheduler. It does not check the requirements for other components, such as DB2. To verify the requirements for Installation Manager use the procedure described in “Scanning system prerequisites for Installation Manager” on page 257.

If any of these checks fails, IBM Workload Scheduler performs the following action:

For all the components installed by using Installation Manager:

Displays a notification of the requirement that was not met. In this case, stop the installation or the upgrade, analyze the log files, solve the error, and rerun the installation or upgrade. If you are performing an interactive installation, the errors are displayed on the screen. If you are performing a silent installation, the errors are written in the Installation Manager log files. For more information about log files, see Installation Manager wizard, silent installation and uninstallation log files in *IBM Workload Scheduler: Planning and Installation*.

For agents

An error message is returned. In this case, analyze the log file, solve the error, and rerun the installation or upgrade. The log files are located:

On Windows operating systems:

```
%TEMP%\TWA\tws93\result.txt
```

On UNIX and Linux operating systems:

```
$tmp/TWA/tws93/result.txt
```

You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you specify the **-skipcheckprereq** parameter, the **twsinst** installation script does not execute the prerequisite scan. If a problem occurs, an error is displayed, the agent is installed or upgraded, but might not work. For more information about the **-skipcheckprereq** option, see Agent installation parameters in *IBM Workload Scheduler: Planning and Installation*.

For a detailed list of supported operating systems and product prerequisites, see the System Requirements Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

Scanning system prerequisites for Installation Manager

Scanning system prerequisites for Installation Manager before you install or upgrade the IBM Workload Scheduler.

About this task

Before you install or upgrade the IBM Workload Scheduler, if you have not installed Installation Manager, run a scan on your system to verify that your workstation has all the system requirements needed for a successful installation. Having an environment that meets the product system requirements ensures that an installation succeeds without any delays or complications.

You can run a prerequisite scan for Installation Manager by using **checkPrereq** command:

On Windows operating systems:

Run the following command:

```
checkPrereq.bat
```

On UNIX or Linux operating systems:

Run the following command:

```
checkPrereq.sh
```

Specify the **-silent** option if you are not interested in installing Installation Manager by using the wizard. If you use the **-silent** option, the program does not check that the graphical libraries exist. If the scan fails, the program displays a notification of the requirement that was not met. In this case, stop the installation, solve the error, and rerun the installation.

Chapter 13. Installing

Installing the Dynamic Workload Console.

Install this component if you want to manage your static and dynamic workload both in distributed and end-to-end environments using a web interface.

By default the Dynamic Workload Console installation process installs the z/OS connector component. During installation process, you might also configure the z/OS connector instance to connect to the z/OS system.

Selecting your installation method

You can install the Dynamic Workload Console using one of the following methods:

Launchpad

Use the launchpad to guide you through the installation of the Dynamic Workload Console, and the IBM Workload Scheduler components, from a single interface. For more information about how to install using the launchpad, see “Installation procedure for Dynamic Workload Console.”

Installation wizard

Install the Dynamic Workload Console and the z/OS connector by using the Installation Manager wizard for each supported platform. The wizard guides you through the installation steps.

For more information, see “Installation procedure for Dynamic Workload Console.”

Silent mode

Using this method, you run the installation unattended and in the background. A response file provides the relevant information to the installation process, which is run in background. Customize the *response file* by adding all the configuration settings to be used during the installation. Then, from the command line, run the Installation Manager command. For more information, see “Performing a silent installation” on page 264.

Installation procedure for Dynamic Workload Console

About this task

To install a Dynamic Workload Console and its prerequisites, perform the following steps:

1. Run the installation process:

Launchpad

To start the launchpad installation program, perform the following steps:

- a. From the DVD that contains the IBM Workload Scheduler master domain manager package to install, run the launchpad as follows:

Windows operating systems:

From the root directory of the DVD, run `launchpad.exe`.

UNIX operating systems:

- 1) Export the browser location to the BROWSER environment variable.
- 2) From the root directory of the DVD, run `launchpad.sh`.

The launchpad opens.

- b. In the left frame of the launchpad, click the following statement **Installing > DWC and z/OS Connector**. Select the IBM Workload Scheduler hyperlink. The Installation Packages Installation Manager panel opens.

Installation wizard

To start the installation program, perform the following steps:

- a. From the eImage or the DVD that contains the IBM Workload Scheduler master domain manager, run:

Windows operating systems:

From the root directory of the DVD or the eImage, run `setupDWC.bat`.

UNIX operating systems:

With IBM Workload Scheduler version 9.3 you are no longer required to have root privileges to install the Dynamic Workload Console.

If you decided to silently install the Dynamic Workload Console as non-root user, also the Installation Manager must be installed by the same non-root user.

If you are installing as the root user

From the root directory of the DVD or the eImage, run `setupDWC.sh`.

If you are installing as a non-root user

Complete the following steps:

- 1) Create the `/etc/TWA` folder and grant write access to the non-root user that is installing the Dynamic Workload Console.
- 2) Assign to the `<DWC_INSTALL_MEDIA>` directory the non-root user ownership, by running the following command:

```
chown -R non-root-user /<DWC_INSTALL_MEDIA>
```
- 3) Run `setupDWC.sh -noroot`

Note: If you are installing the Dynamic Workload Console as a non-root user, consider the following limitations:

- All the prerequisite products, as listed in Chapter 12, “Dynamic Workload Console prerequisites,” on page 255, must be installed by the non-root user.
- After the installation, fix packs can be installed only by the non-root user.
- Only fix packs and releases certified for the non-root user can be applied on that Dynamic Workload Console instance.

- The default directory is /opt/IBM/TWAUI

Note: If you are installing on a Windows server 2008 follow the instructions in the message about virtualized directories.

5. Click **Next**. On the Features page, select the languages for which the corresponding WebSphere Application Server packages will be installed. The language translations for the user interface and documentation are installed. You have the option to select languages only the first time that you install a package to a package group. You can install other language translations for all the packages in a package group with the Modify wizard. Click **Next**.
6. On the Features page, perform the following actions:

For the prerequisite packages:

To see a description of a feature, click the feature name. In the Details section you see a short description.

Ensure that you leave the default prerequisites features selected by installation process.

For the Dynamic Workload Console package:

Leave the **Dynamic Workload Console** option selected.

Click **Next**.

7. In the following panels, enter the following information:

For each prerequisite package:

On the prerequisites product panels, enter the information related to the product you are installing. For more information about the field values, see the prerequisite product documentation.

For the Dynamic Workload Console package:

On the following panels, enter the following information:

WebSphere Application Server profile configuration:

“WebSphere Application Server profile configuration” on page 263

z/OS connector configuration:

“z/OS connector configuration” on page 264.

Click **Next**.

8. On the Summary page, review your choices before installing the product package and its prerequisites. To change any choices that you made on previous pages, click **Back** and make the changes. Click **Install** to install the Dynamic Workload Console package and its prerequisites.

Note: If you installed the WebSphere Application Server prerequisite, after the installation do not create a profile because the installation process already created its own profile.

After a successful installation, to configure the Dynamic Workload Console, see “Accessing the Dynamic Workload Console” on page 272.

WebSphere Application Server profile configuration

About this task

The following fields are provided for WebSphere Application Server profile configuration data. The fields you complete depend upon whether you are using an existing profile for Dashboard Application Services Hub or another profile.

WebSphere installation location

Type or **Browse** for the directory where the WebSphere Application Server instance is installed. Click **Browse** to find the appropriate location.

Use an existing WebSphere Application Server profile

You use a WebSphere Application Server profile that you have already created.

Profile details

Profile location

Enter the name of the directory where the WebSphere Application Server profile is located. Click **Browse**, to find the appropriate location. The default is:

On Windows operating systems:

C:\Program Files\IBM\JazzSM\profile

On UNIX operating systems:

/opt/IBM/JazzSM/profile

Note: Do not use any of the following characters in the profile path field:

On Windows operating systems:

!"#\$%&{}[]=?'<>,,*:

On UNIX operating systems:

!"#\$%&{}[]=?'<>,,*:

Profile name

Enter the name of the file where the WebSphere Application Server profile is defined. The default is **JazzSMProfile**.

Node name

Enter the name of the node contained in the WebSphere Application Server profile. The default is **JazzSMNode01**.

Server name

Enter the name of the server contained in the WebSphere Application Server profile. The default is **server1**.

User name

Provide the user that can access the WebSphere Application Server profile. The default is **wasadmin**.

Password

Provide the WebSphere Application Server password for the user you specified. The password must comply with the password policy in your Local Security settings.

Validate

Click validate the information you entered are correct.

z/OS connector configuration

About this task

Specify the information to connect to an IBM Workload Scheduler for z/OS system.

Configure a connection to an IBM Workload Scheduler for z/OS host

Select it if you want to create a connection to an IBM Workload Scheduler for z/OS controller.

IBM Workload Scheduler for z/OS engine name

Specify the name of the IBM Workload Scheduler for z/OS engine which you are connecting to.

IBM Workload Scheduler for z/OS remote host

Specify the host name or TCP/IP address of the remote z/OS system where the IBM Workload Scheduler for z/OS controller is installed.

IBM Workload Scheduler for z/OS remote TCP/IP port

Specify the number of the TCP/IP port of the z/OS system used to communicate with the IBM Workload Scheduler for z/OS controller.

Enable SSL

Select to enable the SSL communication between the z/OS connector and the remote z/OS system. By default, this box is not selected and the communication is not SSL.

Performing a silent installation

About this task

When you run a silent installation, you have the Installation Manager already installed and you use an XML response file that contains parameters required to install the product package.

The response file includes all the installation information required to run the installation without user intervention.

To silently install Dynamic Workload Console product package you can have one the following scenarios:

Installing the Dynamic Workload Console package:

The Dynamic Workload Console prerequisites are already installed. For more information about performing the silent installation of Dynamic Workload Console package, see "Performing a Dynamic Workload Console silent installation."

Installing the Dynamic Workload Console package and its prerequisites:

For more information about performing the silent installation of the Dynamic Workload Console prerequisites packages and its prerequisites , see "Performing a Dynamic Workload Console and its prerequisites silent installation" on page 266.

Performing a Dynamic Workload Console silent installation

Before you begin

You must install Installation Manager before you perform a silent installation of the Dynamic Workload Console package.

If you decided to silently install the Dynamic Workload Console as non-root user, also the Installation Manager must be installed by the same non-root user.

For detailed information about how to install Installation Manager, see the Installation Manager documentation.

About this task

To perform a silent installation of the Dynamic Workload Console package, by using one of the response files listed in “Dynamic Workload Console response file templates” on page 266, perform the following steps:

1. Copy the relevant response file to a local directory.
2. Edit the Dynamic Workload Console section. For details about the response file properties, see “The Dynamic Workload Console response file properties” on page 395.

Note: Ensure that all the passwords that you specify in the response file are encrypted as described in “Encrypting user passwords for response files” on page 268.

3. Save the file with your changes.
4. Open a command-line prompt.
5. Go to the Installation Manager tools directory.

The default tools directory is:

On Windows operating systems

```
C:\Program Files\IBM\Installation Manager\eclipse\tools
```

On UNIX and Linux operating systems

If you are installing as the root user

```
/opt/IBM/InstallationManager/eclipse/tools
```

If you are installing as a non-root user

```
/home/<non-root-user>/IBM/InstallationManager/eclipse/tools
```

6. Run the following command:

On Windows operating systems

```
imcl.exe input <local_dir>\response_file.xml  
-log <local_dir>\log_file.xml  
-acceptLicense
```

On UNIX and Linux operating systems

```
./imcl input /<local_dir>/response_file.xml  
-log /<local_dir>/log_file.xml  
-acceptLicense
```

where

- The `response_file.xml` is the name of the response file to be used for installation.
- The `log_file` is the name of the log file that records the result of the silent installation execution.

Note: For more information about Installation Manager, see Installation Manager documentation.

After a successful installation, perform the configuration tasks as described in the *Dynamic Workload Console User's Guide*.

Dynamic Workload Console response file templates

About this task

Edit the response file templates provided on the installation DVDs in the \response_files\ directory. Instructions for customizing the files are included in the files as commented text.

Table 43 lists the response files and the types of installation each performs by platform:

Table 43. Installation response files

Type of installation	Response file to use
Installing on Windows operating systems	
Fresh Dynamic Workload Console	IWS93_FRESH_DWC_WIN.xml
Installing on UNIX operating systems	
Fresh Dynamic Workload Console	IWS93_FRESH_DWC_UNIX.xml

Table 44 lists the response files and the types of upgrade each performs by platform:

Table 44. Upgrade response files

Type of upgrade	Response file to use
Upgrading on Windows operating systems	
Dynamic Workload Console	TWS93_UPGRADE_DWC_WIN.xml
Upgrading on UNIX operating systems	
Dynamic Workload Console	TWS93_UPGRADE_DWC_UNIX.xml

For details about response file properties, see “The Dynamic Workload Console response file properties” on page 395.

Note: Remember not to modify the following lines at the end of the response file. Be sure that the response file contains the correct component value you want to uninstall in the feature property:

```
<install modify='false'>
<offering id='com.ibm.tws.tdwc'
          profile='Tivoli Dynamic Workload Console'
          features='tws.tdwc' installFixes='none'/>
</install>
```

Performing a Dynamic Workload Console and its prerequisites silent installation

Before you begin

You must install Installation Manager before you perform a silent installation of the Dynamic Workload Console package and its prerequisites.

For detailed information about how to install Installation Manager, see the Installation Manager documentation.

About this task

You can silently install the Dynamic Workload Console package at the same time as its prerequisites packages, by using a global response file that is provided on the installation DVDs in the `\response_files\` directory. For a list of response files, see Table 45 on page 268.

The silent installation process:

- Assigns the correct order to the package installation.
- Manages the prerequisites package installation.

The response file contains one section for each prerequisite package that you have to install and one section related to the Dynamic Workload Console package installation.

Perform the following steps:

1. Copy the response file to a local directory.
2. Edit the following sections:

Dynamic Workload Console prerequisites packages sections:

For more information about how to type in this section see to the response file properties description provided as commented text or see the prerequisites product documentation.

Dynamic Workload Console section:

For more information about how to complete the Dynamic Workload Console section properties, see “The Dynamic Workload Console response file properties” on page 395.

Note: Ensure that all the passwords that you specify in the response file are encrypted as described in “Encrypting user passwords for response files” on page 268.

3. Save the file with your changes.
4. Open a command-line prompt.
5. Go to the Installation Manager tools directory.

The default tools directory is:

On Windows operating systems

`C:\Program Files\IBM\Installation Manager\eclipse\tools`

On UNIX and Linux operating systems

`/opt/IBM/InstallationManager/eclipse/tools`

6. Run the following command:

On Windows operating systems:

```
imcl.exe input <local_dir>\response_file.xml
-log <local_dir>\log_file.xml
-acceptLicense
```

On UNIX and Linux operating systems:

```
./imcl input /<local_dir>/response_file.xml
-log /<local_dir>/log_file.xml
-acceptLicense
```

Where:

- The `response_file.xml` is the name of the response file to be used for installation.

- The `log_file` is the name of the log file that records the result of the silent installation.

Note: For more information about Installation Manager silent installation command and Installation Manager silent log files, see the Installation Manager information center.

Table 45 lists the response files and the types of installation each performs by platform:

Table 45. Global installation response files

Type of installation	Response file to use
Installing on UNIX	
WebSphere Application Server, Jazz for Service Management extension for WebSphere, and Dynamic Workload Console.	TWS93_FRESH_FULL_DWC_UNIX.xml
Installing on Windows	
WebSphere Application Server, Jazz for Service Management extension for WebSphere, and Dynamic Workload Console.	TWS93_FRESH_FULL_DWC_WIN.xml

Encrypting user passwords for response files

Steps that need to be taken to encrypt the user passwords for the response files.

About this task

You must encrypt each password string stored in the response files by using Installation Manager.

You can perform the password encryption by using one of the following procedures:

Installation Manager String encryption utility interface

To encrypt the password string for the response files, perform the following steps:

1. Go to the eclipse directory of the Installation Manager installation directory. The default eclipse directory is:

Windows operating systems

`C:\Program Files\IBM\Installation Manager\eclipse`

UNIX and Linux operating systems

If you are installing as the root user

`/opt/IBM/InstallationManager/eclipse/tools`

If you are installing as a non-root user

`/home/<non-root-user>/IBM/InstallationManager/eclipse/tools`

2. To open the **String encryption utility interface**, run the following command:

Windows operating systems

`IBMIM.exe encryptString <stringToEncrypt>`

where <stringToEncrypt> is the value to be encrypted.

UNIX and Linux operating systems

```
./IBMIM encryptString <stringToEncrypt>
```

where <stringToEncrypt> is the value to be encrypted.

3. In the **String encryption utility** window, note the Encrypted version of the String field value related to the String to be encrypt field value.
4. Copy the Encrypted version of the String value in the password entry of the response file.

Installation Manager command line tool

To encrypt the password string for the response files, perform the following steps:

1. Go to the eclipse directory of the Installation Manager installation directory. The default eclipse directory is:

Windows operating systems

```
C:\Program Files\IBM\Installation Manager\eclipse
```

UNIX and Linux operating systems

If you are installing as the root user

```
/opt/IBM/InstallationManager/eclipse/tools
```

If you are installing as a non-root user

```
/home/<non-root-user>/IBM/InstallationManager/eclipse/  
tools
```

2. Run the following command:

Windows operating systems

```
IBMIM.exe -silent -noSplash encryptString <stringToEncrypt> >  
<Encryptedpwd>.txt
```

where <stringToEncrypt> is the value to be encrypted and the <Encryptedpwd>.txt is the file where there is the encrypted value of the password.

UNIX and Linux operating systems

```
./IBMIM -silent -noSplash encryptString <stringToEncrypt> >  
<Encryptedpwd>
```

where <stringToEncrypt> is the value that is encrypted and the <Encryptedpwd> is the file where there is the encrypted value of the password.

3. Open the file <Encryptedpwd> and copy the value contained into the file in the data key of the response file.
4. Remove the file <Encryptedpwd>.

Example

This example shows you how to write the section USER INFORMATION of the IWS93_FRESH_MDM_WIN.xml response file, setting the IBM Workload Scheduler user value to *twsuser* and the user password value to *password* on Windows operating systems.

By using the Installation Manager command line tool, encrypt the password *passwd0rd* saving the encrypted value to the file *my_pwd.txt*:

```
IBMIM.exe -silent -noSplash encryptString passwd0rd > my_pwd.txt
```

The file *my_pwd.txt* contains the following value:

```
rbN1IaMAWYYtQxLf6KdNyA==
```

Complete the USER INFORMATION section of the *IWS93_FRESH_MDM_WIN.xml* response file as follows:

```
<!--USER INFORMATION
Supply the IBM Workload Scheduler credentials information -->
<data key='user.userName,com.ibm.tws' value='twsuser' />
<data key='user.password,com.ibm.tws' value='rbN1IaMAWYYtQxLf6KdNyA==' />
```

Note: For security reasons, remove the file *my_pwd.txt* after using it.

Installation log files

Log files created by the installation process.

For more information about log files, see the *Administration Guide*.

Interactive wizard installation and uninstallation log files

Installation Manager creates the following installation and uninstallation logs files common to any package installation, regardless of which components you choose to install:

On Windows operating systems

```
<INSTALLATION_MANAGER_LOGS_DIR>\<YYYYMMDD_HHMM>.xml
```

On UNIX and Linux operating systems

```
<INSTALLATION_MANAGER_LOGS_DIR>/<YYYYMMDD_HHMM>.xml
```

where *<INSTALLATION_MANAGER_LOGS_DIR>* is the directory where Installation Manager creates the log files, *YYYYMMDD* is the date and *HHMM* is the time when the log file is created.

The Dynamic Workload Console installation process creates the following Installation Manager native logs files:

On Windows operating systems

```
<INSTALLATION_MANAGER_LOGS_DIR>\native\<YYYYMMDD_HHMM>.log
```

On UNIX and Linux operating systems

```
<INSTALLATION_MANAGER_LOGS_DIR>/native/<YYYYMMDD_HHMM>.log
```

where *<INSTALLATION_MANAGER_LOGS_DIR>* is the directory where Installation Manager creates the logs files, and *YYYYMMDD* is the date and *HHMM* is the time when the log file is created.

The *<INSTALLATION_MANAGER_LOGS_DIR>* default is:

On Windows operating systems

```
C:\ProgramData\IBM\InstallationManager\logs
```

On UNIX and Linux operating systems

```
/var/ibm/InstallationManager/logs
```

If more than one native log have the same timestamp, Installation Manager creates the log files with the following name:

On Windows operating systems

<INSTALLATION_MANAGER_LOGS_DIR>\native\<YYYYMMDD_HHMMLETTER>.log

On UNIX and Linux operating systems

<INSTALLATION_MANAGER_LOGS_DIR>/native/<YYYYMMDD_HHMMLETTER>.log

where <INSTALLATION_MANAGER_LOGS_DIR> is the directory where Installation Manager creates the logs files, YYYYMMDD is the date, HHMM is the time when the log file is created, and LETTER is a letter of the alphabet.

For more information about how to access the log files by using the Installation Manager wizard, see “Accessing Installation Manager log files via wizard.”

For more information about how to create a .zip file of the native log directory, see “Packaging Installation Manager log files via wizard”

Accessing Installation Manager log files via wizard

About this task

By using the Installation Manager wizard, you can access the Installation Manager log files in the following log directory:

<INSTALLATION_MANAGER_LOGS_DIR>

where <INSTALLATION_MANAGER_LOGS_DIR> is the directory where Installation Manager creates the logs files. The <INSTALLATION_MANAGER_LOGS_DIR> default value is:

On Windows operating systems

C:\ProgramData\IBM\InstallationManager\logs

On UNIX and Linux operating systems

/var/ibm/InstallationManager/logs

To access the log files by using the wizard, perform the following steps:

1. Open the Installation Manager Start page.
2. Select **File>View Log**.
3. The Installation Log panel shows you all the log files saved on your machine. Select the log file whose name is the correct timestamp for your installation process.
4. Depending on the action that you want to perform, click the **Export log file** icon or **Open log file** icon on the upper right side.

Packaging Installation Manager log files via wizard

About this task

By using the Installation Manager wizard, you can create a .zip file that contains the following log files:

- Native log files in the <INSTALLATION_MANAGER_LOGS_DIR>\native directory.
- xml log files in the <INSTALLATION_MANAGER_LOGS_DIR> directory.

Where <INSTALLATION_MANAGER_LOGS_DIR> is the directory where Installation Manager creates the log files.

The <INSTALLATION_MANAGER_LOGS_DIR> default value is:

On Windows operating systems

C:\ProgramData\IBM\InstallationManager\logs

On UNIX and Linux operating systems

/var/ibm/InstallationManager/logs

To create a *.zip* file of the *native* log directory, perform the following steps:

1. Open the Installation Manager Start page.
2. Select **Help>Export Data for Problem Analysis**.
3. Enter the name of the directory where you want to create the *.zip* file and the *.zip* file name.
4. Press **Ok**. A *.zip* file that contains all log files is created in the directory you specified.

Jazz for Service Management extension for WebSphere profile log files

The Dynamic Workload Console installation process manages its own profile in the Jazz for Service Management extension for WebSphere instance.

The log for the Jazz for Service Management extension for WebSphere profile management of the application server can be found at:

/<JAZZ_SM_HOME>/profile/logs/<SERVER_NAME>

where <JAZZ_SM_HOME> is the Jazz for Service Management extension for WebSphere installation directory and the <SERVER_NAME> is the server name related to the Jazz for Service Management extension for WebSphere that you use.

Accessing the Dynamic Workload Console

From a supported browser, access one of the following links provided by the installation program:

http://dynamic_workload_console_system:http_port/DASH_context_root

https://dynamic_workload_console_system:https_port/DASH_context_root

where:

dynamic_workload_console_system

The hostname or IP address of the system where you installed the Dynamic Workload Console.

http_port

The port number used to access the Dynamic Workload Console using an unsecure connection over HTTP. The default value for this port number is **16310**.

https_port

The port number used to access the Dynamic Workload Console using a secure connection over HTTPS. The default value for this port number is **16311**.

When connecting to the Dashboard Application Services Hub using an HTTPS connection, if you receive a security alert, proceed with the Dynamic Workload Console working session. If you receive security

information windows while navigating through the Dashboard Application Services Hub, choose to display nonsecure items to proceed. If you are using Internet Explorer, you can prevent these windows from opening by setting **Display mixed content** to **Enable** in the Security settings.

DASH_context_root

It is the Dashboard Application Services Hub context root defined at installation time. The context root determines the URL of a deployed application and by default is identical with the application directory or archive structure. In this case, the default is `ibm/console`.

In the Dashboard Application Services Hub login portlet, enter the user ID and password you specified during the installation, and click **Log in**.

Access Single Entry Pointpage and click Go from the Dynamic Workload Console launching point.

For a quick and rapid overview of the portal and of its use, after logging in to the Dashboard Application Services Hub, click one of the hyperlinks displayed on the welcome page to launch videos and a tutorial that help you find the information you need.

Several products might be integrated in this portal and their related entries are listed together with those belonging to the Dynamic Workload Console in the toolbar. Use these icons to perform your tasks.

Use the toolbar to work with the Dynamic Workload Console to perform the IBM Workload Scheduler tasks.

To effectively use the functions of the IBM Workload Scheduler and the dynamic workload broker, you must define connections to the IBM Workload Scheduler engines and the dynamic workload broker servers.

If you do not define engine connections, you can perform only this limited set of operations:

On IBM Workload Scheduler:

- Create monitor tasks
- Create report tasks
- Create event management tasks
- Define user preferences

On dynamic workload broker

Define user preferences

If the user ID you used to connect to the Dynamic Workload Console has been assigned a role different from **TWSWEBUIAdministrator** and **TDWBAdministrator**, you will see a subset of the available panels. This subset depends on the authorizations assigned to the role associated to your user ID. For more information about roles, see the information about configuring the Dynamic Workload Console in the *IBM Workload Scheduler: Administration Guide*.

If the user ID you used to connect to the Dynamic Workload Console has no role assigned, you do not see the entries for IBM Workload Scheduler and dynamic workload broker in the Dashboard Application Services Hub navigation tree.

Quick steps to define an IBM Workload Scheduler engine connection

Steps to create an engine connection to one of your supported IBM Workload Scheduler engines.

After logging in to the Dynamic Workload Console using the administrator user ID or another user ID with assigned **TWSWEBUIAdministrator** or **TWSWEBUIConfigurator** roles, use the following steps to create an engine connection to one of your supported IBM Workload Scheduler engines.

1. From the navigation toolbar, click **System Configuration > Manage Engines**.
2. From the displayed panel you can create, edit, delete, or share an engine connection, and test the connection to the remote server where IBM Workload Scheduler is installed. You can order the list of engine connections displayed in this panel by using sorting criteria that you select with the buttons at the upper left corner of the table.
3. Click **New Engine**.
4. In the Engine Connection Properties window, assign a name to the engine connection and specify the required information. For more details about fields and options, see the online help by clicking the "?" in the top right corner. If you want to test the connection to the IBM Workload Scheduler database (mandatory for managing reporting and event management functions), you must select **Enable reporting** and specify the user credentials.
5. Click **Test Connection** to check that the configuration was successful and that the Dynamic Workload Console is communicating with the selected engine. If the test connection fails, see *IBM Workload Scheduler: Troubleshooting Guide*.

Quick steps to define an dynamic workload broker connection

Steps to take to create a connection to a supported dynamic workload broker engine.

The Dynamic Workload Console supports a single connection to one dynamic workload broker engine at any given time for each authorized user. A different connection is supported for each authorized user.

After logging in to the Dynamic Workload Console using the administrator user ID, or another user ID with assigned **TDWBAdministrator** or **TDWBConfigurator** roles, follow these steps to create an engine connection to a supported dynamic workload broker engine:

1. In the Dynamic Workload Console, click **dynamic workload broker** to expand the tree.
2. Select **Configuration**.
3. Click **Server connection**.
4. In the Server Connection specify:

Hostname

The host name of the dynamic workload broker you want to connect to.

Non secure port

The non-secure port to be used for connection.

Secure port

The secure port to be used for connection.

Use Secure Connection

Specify whether a secure connection must be used. For more information about security, see the *IBM Workload Scheduler: Administration Guide*.

Username

Optionally specify a different user for the server connection. The connection to the new server is enabled using the credentials of the user you specified. Each user has access to only one server connection.

Password

Specify the password for the authenticated user the connection applies to.

5. Click **OK** to save your changes. The server connection you specified is enabled and is immediately effective.

Starting and stopping the Dynamic Workload Console

Options on starting and stopping the Dynamic Workload Console.

To start and stop the Dynamic Workload Console, related to a Jazz for Service Management extension for WebSphere profile, you must start and stop the WebSphere Application Server instance by using one of the following options:

wastools installed for the Dynamic Workload Console:

To Start the Dynamic Workload Console:

On Windows operating systems:

```
<DWC_INST_DIR>\wastools\startWas.bat
```

On UNIX and Linux operating systems

```
<DWC_INST_DIR>/wastools/startWas.sh
```

To Stop the Dynamic Workload Console:

On Windows operating systems

```
<DWC_INST_DIR>\wastools\stopWas.bat
```

On UNIX and Linux operating systems:

```
<DWC_INST_DIR>/wastools/stopWas.sh
```

where <DWC_INST_DIR> is the Dynamic Workload Console installation directory.

For more information about the utilities usage, see *Administration Guide: Application server tasks*.

Note: When you start or stop the Dynamic Workload Console, related to a Jazz for Service Management extension for WebSphere profile, you are prompted to insert the credentials for the profile. To avoid this behaviour, run the command with `-direct` option.

WebSphere Application Server native commands:

To Start the Dynamic Workload Console:

On Windows operating systems:

```
<JAZZSM_INSTALL_DIR>\profile\bin\startServer.bat  
<app_server>
```

On UNIX and Linux operating systems:

```
<JAZZSM_INSTALL_DIR>/profile/bin/startServer.sh  
<app_server>
```

To Stop the Dynamic Workload Console:

On Windows operating systems:

```
<JAZZSM_INSTALL_DIR>\profile\bin\stopServer.bat  
<app_server>  
-user <user_id> -password <user_id_pw>
```

On UNIX and Linux operating systems:

```
<JAZZSM_INSTALL_DIR>/profile/bin/stopServer.sh  
<app_server>  
-user <user_id> -password <user_id_pw>
```

where:

<JAZZSM_INSTALL_DIR>

Is the directory where the Jazz for Service Management extension for WebSphere is installed.

<app_server>

Is the server name specified in the Jazz for Service Management extension for WebSphere profile related to the Dynamic Workload Console. The default is **server1**.

<user_id>

Is the administrator user ID specified when installing the Dynamic Workload Console.

<user_id_pw>

Is the administrator user ID password specified when installing the Dynamic Workload Console.

Chapter 14. Configuring

Some links and pointers on configuration tasks that are needed for the Dynamic Workload Console.

The following is a list of links or pointers to places that document the configuration tasks needed for the Dynamic Workload Console. You can perform the following optional configuration steps at any time after the installation.

- Configuring new users to access the Dynamic Workload Console: see the section about configuring access to the Dynamic Workload Console in the *IBM Workload Scheduler: Administration Guide*.
- Configuring the Dynamic Workload Console to use a user registry:
 - For configuring the Dynamic Workload Console with LDAP - RACF, see the WebSphere documentation at: Configuring to secure Lightweight Directory Access Protocol user registry using Resource Access Control Facility based on z/OS.
Also, see “Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP)” on page 278.
 - For configuring access to the Dynamic Workload Console, see the corresponding section in the *IBM Workload Scheduler: Administration Guide*.
- Configuring roles to access the Dynamic Workload Console: see the corresponding section in the *IBM Workload Scheduler: Administration Guide*.
- Configuring the Dynamic Workload Console to use Single Sign-On: see the corresponding section in the *IBM Workload Scheduler: Administration Guide*.
- Securing your communication with the Secure Socket Layer protocol: see the section about customizing the SSL connection between the Dynamic Workload Console and components with a distributed connector in the *IBM Workload Scheduler: Administration Guide*.
- Creating a Windows service for Jazz for Service Management extension for WebSphere: see the section in the *IBM Workload Scheduler: Administration Guide*.
- Configuring the Dynamic Workload Console to launch in context: see the corresponding section in the *IBM Workload Scheduler: Administration Guide*.

Note: If, after installing, you have more than one instance of WebSphere Application Server managing any IBM Workload Automation products, you must ensure that they have the same LTPA token_keys.

For detailed information about how to configure the Dynamic Workload Console, see the *IBM Workload Scheduler: Administration Guide*.

For more information about configuring authentication using the Lightweight Directory Access Protocol (LDAP), see the *IBM Workload Scheduler: Administration Guide*.

Post-installation steps to configure the use of Lightweight Third-Party Authentication (LDAP)

If the Dynamic Workload Console and the IBM Workload Scheduler engine or the IBM Workload Scheduler z/OS Connector have been configured with the same LDAP user registry, or are installed on the same computer, you might receive a connection failure. If this happens, use the same Lightweight Third-Party Authentication (LTPA) keys on all servers: the Dynamic Workload Console, the IBM Workload Scheduler engine server, and the IBM Workload Scheduler z/OS Connector server.

To align the LTPA keys, see [Configuring the use of Lightweight Third-Party Authentication](#) in the IBM Workload Scheduler Administration manual.

Chapter 15. Navigating the Dynamic Workload Console

An overview to the Dynamic Workload Console.

For an interactive overview of the product and its features, you can view several demo scenarios, available (in English only) on the IBM Workload Automation YouTube channel.

To have a quick and rapid overview of the portal and of its use, after logging in, the Welcome page for the Dynamic Workload Console is displayed in the Dashboard Application Services Hub console window. This window has a navigation menu across the top, organized in categories. Each category drops down to display a number of options that when clicked, display a portlet in the work area on the right. Each portlet displays with a title in its tabbed window in the work area. Just as the navigation menu items are customized according to the role of the logged in user, the welcome page is also customized for the user. The **Quick start** tasks available on the **Welcome** page allow you to access a related how-to video and launch the related portlet. Only the tasks corresponding to the logged in user's role are displayed. To get oriented with the navigation bar, take the tour and explore the available items. The tour brings into focus each of the navigation bar categories and corresponding descriptive text is displayed. There are other helpful links on the page such a link to the embedded online help, the IBM Workload Automation YouTube channel, and a QR code to scan with your mobile device to launch the mobile applications.

Several products can be integrated in this portal and their related entries are listed together with those belonging to the Dynamic Workload Console in the navigation bar displayed at the top of the page.

The navigation bar at the top of the page is your entry point to the Dynamic Workload Console.

First actions

Some first actions that need to be performed when connecting to the Dynamic Workload Console .

The first and main actions you perform when you connect to the Dynamic Workload Console.

Creating a connection to an IBM Workload Scheduler engine

You type the details (such as IP address, user name, and password) to access an IBM Workload Scheduler engine, and, optionally, a database to operate with objects defined in plans or stored in the database. From the Dynamic Workload Console you can access the current plan, a trial plan, a forecast plan, or an archived plan for the distributed environment or the current plan for the z/OS® environment. You might want to access the database to perform actions against objects stored in it or generate reports showing historical or statistical data. In addition, working both on the database and on plans, you can create and run event rules to define and trigger actions that you want to run in response to events occurring on IBM Workload Scheduler nodes.

Defining a scheduling environment

You define your IBM Workload Scheduler network. You create workstation definitions on the database representing the physical machines or computer systems on which your workload is scheduled to run. The IBM Workload Scheduler network is made up of the workstations where job and job stream processing occurs. When you design your network, you assign roles to these workstations to suit your specific business requirements. You can design your network with multiple domains, to divide control of a large network into smaller manageable groups. A typical IBM Workload Scheduler network consists of a workstation acting as a master domain manager and at least one domain. See Dynamic Workload Console User's Guide, section about Creating and managing engine connections.

Defining scheduling objects in the database

You define your workload, which consists of jobs that are concatenated in job streams. Then, you specify the calendars and run cycles according to which job streams must run. Moreover, you define possible dependencies to condition the workload processing. All these definitions can be done within the Workload Designer. See Dynamic Workload Console User's Guide, section about Designing your Workload.

Creating tasks to manage IBM Workload Scheduler objects in the plan

You specify some filtering criteria to query a list of scheduling objects whose attributes satisfy the criteria you specified. Starting from this list, you can navigate and modify the content of the plan, switching between objects, opening more lists, and accessing other plans or other IBM Workload Scheduler environments. See Dynamic Workload Console User's Guide, section about Monitoring your Workload.

Creating a connection to a Tivoli dynamic workload broker scheduling environment

You type the details (such as IP address, user name, password, and port) to access a dynamic workload broker workstation. Specify if you want to work in a secure HTTPS or HTTP protocol. After creating the connection, by opening the tracking computer you can view status and details of broker workstations, and define resources and dynamic jobs. For more details about dynamic scheduling, see IBM Workload Scheduler Scheduling Workload Dynamically.

Chapter 16. Upgrading

Upgrading the Dynamic Workload Console.

This chapter describes how to upgrade Dynamic Workload Console from version 8.6.0 and later to the current version.

Upgrading overview

Upgrading the Dynamic Workload Console from version 8.6.0 and later to the current version.

This section provides an overview of the upgrade process of an existing version of Dynamic Workload Console V8.6.0 and later instances.

Installation process changes

The following changes from V9.1 affect the installation process:

- External WebSphere Application Server prerequisite.
- Jazz for Service Management extension for WebSphere prerequisite.
- Dashboard Application Services Hub prerequisite.

Upgrade deploy model for single or multiple V8.6 component instances installed in the directory <TWS_INST_DIR>:

Single instance:

A *single instance* contains the Dynamic Workload Console component installed in the <TWS_INST_DIR> directory.

Multiple instance:

A *multiple instance* contains the Dynamic Workload Console component and one or two IBM Workload Scheduler components installed in the same <TWS_INST_DIR> directory.

You must know if the instance you are upgrading is *single* or *multiple* to understand which procedure you must use to upgrade the Dynamic Workload Console.

Table 46 shows which procedure you must follow when you upgrade single or multiple instances of Dynamic Workload Console.

Table 46. Upgrade single or multiple instances.

Dynamic Workload Console instance installed in the <TWS_INST_DIR> directory:	Upgrade procedure:
single	"Upgrading a Dynamic Workload Console V8.6 single instance" on page 282
multiple	"Upgrading Dynamic Workload Console V8.6 installed with one or more components in the same directory" on page 292.

Upgrade deploy model for single V9.1 component instances

The upgrade model is available for single component instances only. See "Upgrading Dynamic Workload Console V9.1 or later" on page 296.

Upgrading a Dynamic Workload Console V8.6 single instance

Upgrading a Dynamic Workload Console V8.6 single instance.

You can upgrade a single instance of Dynamic Workload Console in one of the following ways:

Procedure to upgrade Dynamic Workload Console V8.6 on the same workstation where the back-level is installed:

“Upgrading a Dynamic Workload Console V8.6 instance on the same workstation.”

Procedure to upgrade Dynamic Workload Console V8.6 on a new workstation:

“Upgrading Dynamic Workload Console V8.6 single instance on a new workstation” on page 288

Note:

You can upgrade a Dynamic Workload Console on a new workstation only if the new workstation has the same operating system type as the old workstation. Ensure that the new workstation has an operating system supported for V9.3. For more information about supported operating systems, see “Supported operating systems” on page 255.

For example, you can upgrade a Dynamic Workload Console installed on a Windows 32bit workstation to a Windows 64bit workstation. You can upgrade a Dynamic Workload Console installed on a Linux workstation to an AIX workstation or to a Solaris workstation or viceversa.

Procedure to upgrade Dynamic Workload Console V9.2:

“Upgrading Dynamic Workload Console V9.1 or later” on page 296

Upgrading a Dynamic Workload Console V8.6 instance on the same workstation

Upgrading Dynamic Workload Console V8.6 instance on the same workstation.

To upgrade a single instance of Dynamic Workload Console on the same workstation where the back level Dynamic Workload Console is installed, run the following steps:

1. Install a new Dynamic Workload Console in the <DWC_INSTALL_DIR> directory, on the system where the back-level Dynamic Workload Console is installed.
For information about Dynamic Workload Console installation, see “Installation procedure for Dynamic Workload Console” on page 259.
2. Ensure that no Dynamic Workload Console user interface is active and that the WebSphere Application Server is up and running.
3. Migrate the data from the back-level to the newly installed Dynamic Workload Console by running the following script:

On Windows operating systems:

```
<DWC_INSTALL_DIR>\TDWC\scripts\tdwclUpgrade.bat
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
```

```
-newaspassword new_password  
-newtwapath new_twa_path  
[-backuppath backup_path]  
[-machinechange false]
```

On UNIX and Linux operating systems:

```
<DWC_INSTALL_DIR>/TDWC/scripts/tdwcUpgrade.sh  
-oldwasuser old_user  
-oldwaspassword old_password  
-oldtwapath old_twa_path  
-newwasuser new_user  
-newaspassword new_password  
-newtwapath new_twa_path  
[-backuppath backup_path]  
[-machinechange false]
```

where:

-oldwasuser *old_user*

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

-oldwaspassword *old_password*

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

-oldtwapath *old_twa_path*

The installation directory where the back-level Dynamic Workload Console is installed.

-newwasuser *new_user*

The Dashboard Application Services Hub administrator user ID.

-newaspassword *new_password*

The password of the Dashboard Application Services Hub administrator.

-newtwapath *new_twa_path*

The installation directory where you want to install the Dynamic Workload Console. By default the installation directory is:

On Windows operating systems:

```
C:\Program Files\IBM\TWAUI
```

On UNIX and Linux operating systems:

```
/opt/IBM/TWAUI
```

-backuppath *backup_path*

The <BACKUP_DIR> backup directory for the upgrade. By default the backup directory is:

On Windows operating systems:

```
<DWC_INSTALL_DIR>\TDWC\tmp\backup
```

On UNIX and Linux operating systems:

```
<DWC_INSTALL_DIR>/TDWC/tmp/backup
```

This directory contains:

- The **tdwcUpgrade** script log file, `upgrade.log`.
- The files containing the following configuration data exported from the back-level Dynamic Workload Console:
 - The embedded WebSphere Application Server profile in the `UpgradeData.zip` file.
 - The embedded WebSphere Application Server profile registry.

- The port settings in the `ports.txt` file.
- The Tivoli Integrated Portal settings.
- The Dynamic Workload Console settings.

This data is then imported into the newly installed Dynamic Workload Console.

- A compressed file named `backup.zip` containing a saved copy of the configuration data of the newly-installed Dynamic Workload Console. This file is used to roll back to the original configuration if the migration script fails while importing the configuration data from the back-level Dynamic Workload Console.

Note: Because the `backup.zip` file is overwritten every time you run the migration script, it might be useful to save a copy of the first `backup.zip` file containing the original configuration.

-machinechange *false*

To upgrade on the same workstation you must specify the `false` value. The default value is `false`.

Note:

The script replaces any customized data in the new Dynamic Workload Console instance, with the data exported from the old Dynamic Workload Console instance.

A result of `Completed` indicates that the script ran successfully and that the data was correctly imported into the newly-installed Dynamic Workload Console.

If the script fails to import the configuration data into the newly-installed Dynamic Workload Console, a rollback is automatically performed, and the original configuration is restored. To double-check that the rollback ran correctly, ensure that you can access the newly-installed Dynamic Workload Console user interface with the user ID and password specified during the installation.

4. The port numbers used by the two instances of the Dynamic Workload Console are different and they are not automatically migrated by the `tdwcUpgrade` script. Run the following steps to migrate the port numbers of the back-level instance to the newly-installed instance:
 - a. Check that the data was correctly migrated from the old Dynamic Workload Console to the newly-installed Dynamic Workload Console.
 - b. Stop the back-level Dynamic Workload Console.
 - c. Uninstall the back-level Dynamic Workload Console.
 - d. Run the following command:

On Windows operating systems:

```
changeHostProperties <BACKUP_DIR>\ports.txt
```

On UNIX and Linux operating systems:

```
changeHostProperties <BACKUP_DIR>/ports.txt
```

where `<BACKUP_DIR>` is the backup directory.

For more information about this command, see `Application server - using the utilities that change the properties`.

Upgrading Dynamic Workload Console V8.5.1 single instance on a new workstation

To upgrade a Dynamic Workload Console installed in the directory <DWC_BACKLEV_INSTALL_DIR> on your old workstation into the directory <DWC_INSTALL_DIR> on the new workstation, run the following steps:

1. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the new workstation where you want to upgrade the Dynamic Workload Console.
2. Install a new Dynamic Workload Console in the <DWC_INSTALL_DIR> directory on the new workstation.

For information about Dynamic Workload Console installation, see “Installation procedure for Dynamic Workload Console” on page 259.

3. Ensure that no Dynamic Workload Console user interface is active and that the WebSphere Application Server is up and running on the workstation where the back-level is installed.
4. From the new workstation perform the following steps:

On Windows operating systems:

Map the network drive <DWC_BACKLEV_INSTALL_DIR> of the old workstation where the back-level Dynamic Workload Console is installed.

On UNIX and Linux operating systems:

Mount in read-write access the remote file system <DWC_BACKLEV_INSTALL_DIR> where the back-level Dynamic Workload Console is installed. If the mount point name on the new workstation is different from the remote file system name, create a symbolic link between the mount point on the new workstation and the remote file system <DWC_BACKLEV_INSTALL_DIR>, the link name value must be <DWC_BACKLEV_INSTALL_DIR>.

5. Migrate the data from the back-level to the newly-installed Dynamic Workload Console by running the following script from the new workstation:

On Windows operating systems:

```
<DWC_INSTALL_DIR>\TDWC\scripts\tdwclUpgrade.bat
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

On UNIX and Linux operating systems:

```
<DWC_INSTALL_DIR>/TDWC/scripts/tdwclUpgrade.sh
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

Where:

- oldwasuser** *old_user*
The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.
- oldwaspassword** *old_password*
The Tivoli Integrated Portal administrator user password that is specified for the back-level Dynamic Workload Console.
- oldtwapath** *old_twa_path*
The installation directory where the back-level Dynamic Workload Console is installed.
- newwasuser** *new_user*
The Dashboard Application Services Hub administrator user ID. The new user ID must be different from the old one.
- newwaspassword** *new_password*
The password of the Dashboard Application Services Hub administrator.
- newtwapath** *new_twa_path*
The installation directory where the Dynamic Workload Console must be installed. By default the installation directory is:
 - On Windows operating systems:**
C:\Program Files\IBM\TWAUI
 - On UNIX and Linux operating systems:**
/opt/IBM/TWAUI
- backuppath** *backup_path*
The <BACKUP_DIR> backup directory for the upgrade process.
By default the installation directory is:
 - On UNIX and Linux operating systems:**
<DWC_INSTALL_DIR>/TDWC/tmp/backup
 - On Windows operating systems:**
<DWC_INSTALL_DIR>\TDWC\tmp\backup

This directory contains:

- The **tdwcUpgrade** script log file, `upgrade.log`.
- The files containing the following configuration data exported from the back-level Dynamic Workload Console:
 - The embedded WebSphere Application Server profile.
 - The embedded WebSphere Application Server profile registry.
 - The Tivoli Integrated Portal settings.
 - The Dynamic Workload Console settings.

This data is then imported into the newly installed Dynamic Workload Console.

- A compressed file named `backup.zip` containing a saved copy of the configuration data of the newly-installed Dynamic Workload Console. This file is used to roll back to the original configuration if the migration script fails while importing the configuration data from the back-level Dynamic Workload Console.

Note: Because the `backup.zip` file is overwritten every time you run the migration script, it might be useful to save a copy of the first `backup.zip` file containing the original configuration.

-machinechange true

You must specify the true value to upgrade the Dynamic Workload Console on the new workstation.

Note: The script replaces any customized data in the new Dynamic Workload Console instance, with the data exported from the old Dynamic Workload Console instance. A result of Completed indicates that the script ran successfully and that the data was correctly imported into the newly-installed Dynamic Workload Console.

If the script fails to import the configuration data into the newly-installed Dynamic Workload Console, a rollback is automatically performed and the original configuration is restored. To double-check that the rollback ran correctly, ensure that you can access the newly-installed Dynamic Workload Console user interface with the user ID and password specified during the installation.

6. Check that the data was correctly migrated from the old Dynamic Workload Console to the newly-installed Dynamic Workload Console.
7. The port numbers used by the two instances of the Dynamic Workload Console might be different, and they are not automatically migrated by the **tdwcUpgrade** script. If the Dynamic Workload Console ports of the instance installed on the new workstation are different from the ports installed on the old machine, and you want to have the same values, run the following steps to migrate the port numbers:
 - a. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the workstation where the back-level Dynamic Workload Console is installed.
 - b. To save the Dynamic Workload Console ports data, redirect the **showHostProperties** script output to the `HostProperties_file` file:

On Windows operating systems:

From the `<DWC_BACKLEV_INSTALL_DIR>\wastools:`

```
showHostProperties.sh  
--username old_DWCuser  
--password old_DWCpassword
```

> `HostProperties_file`

On UNIX and Linux operating systems:

From the `<DWC_BACKLEV_INSTALL_DIR>/wastools:`

```
showHostProperties.bat  
--username old_DWCuser  
--password old_DWCpassword
```

> `HostProperties_file`

where:

old_DWCuser

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

old_DWCpassword

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

- c. Copy the `HostProperties_file` created in step 7b on page 287 in the old workstation to the new workstation in the `<BACKUP_DIR>` backup directory that you used for the upgrade process. By default the backup directory used in the upgrade process is:

On Windows operating systems:

`<DWC_INSTALL_DIR>\TDWC\tmp\backup`

On UNIX and Linux operating systems:

`<DWC_INSTALL_DIR>/TDWC/tmp/backup`

- d. To import the old workstation port values to the Dynamic Workload Console installed on the new workstation, run:

On Windows operating systems:

`changeHostProperties <BACKUP_DIR>\HostProperties_file`

On UNIX and Linux operating systems:

`changeHostProperties <BACKUP_DIR>/HostProperties_file`

For more information about this command, see the section about application server - using the utilities that change the properties in *Administration Guide*.

8. Optionally, uninstall the back-level Dynamic Workload Console on the old machine.

Upgrading Dynamic Workload Console V8.6 single instance on a new workstation

To upgrade a Dynamic Workload Console installed in the directory `<DWC_BACKLEV_INSTALL_DIR>` of your old workstation in the directory `<DWC_INSTALL_DIR>` of the new workstation, run the following steps:

1. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the workstation where the back-level Dynamic Workload Console is installed.
2. To save the back-level Tivoli Integrated Portal profile data, run the **preupgrade** script:

On Windows operating systems:

From `<DWC_BACKLEV_INSTALL_DIR>\eWAS\profiles\TIPProfile\upgrade\bin:`

preupgrade.sh

`--username old_DWCuser`
`--password old_DWCpassword`

On UNIX and Linux operating systems:

From `<DWC_BACKLEV_INSTALL_DIR>/eWAS/profiles/TIPProfile/upgrade/bin:`

preupgrade.bat

`--username old_DWCuser`
`--password old_DWCpassword`

where:

`old_DWCuser`

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

old_DWCpassword

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

Note: The **preupgrade** script creates the following .zip file that contains the back-level Tivoli Integrated Portal profile data:

On Windows operating systems:

<DWC_BACKLEV_INSTALL_DIR>\eWAS\profiles\TIPProfile\upgrade\data\upgradeData.zip:

On UNIX and Linux operating systems:

<DWC_BACKLEV_INSTALL_DIR>/eWAS/profiles/TIPProfile/upgrade/data/upgradeData.zip:

3. To save the Dynamic Workload Console ports data, redirect the **showHostProperties** script output to the `HostProperties_file` file:

On Windows operating systems:

From <DWC_BACKLEV_INSTALL_DIR>\wastools:

showHostProperties.sh

--username *old_DWCuser*

--password *old_DWCpassword*

> HostProperties_file

On UNIX and Linux operating systems:

From <DWC_BACKLEV_INSTALL_DIR>/wastools:

showHostProperties.bat

--username *old_DWCuser*

--password *old_DWCpassword*

> HostProperties_file

where:

old_DWCuser

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

old_DWCpassword

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

4. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the new workstation where you want to upgrade the Dynamic Workload Console.
5. Install a new Dynamic Workload Console in the <DWC_INSTALL_DIR> directory of the new workstation.

For information about Dynamic Workload Console installation, see “Installation procedure for Dynamic Workload Console” on page 259.

Note: When installing the new Dynamic Workload Console instance, the default server name proposed by the installation is **server1**, defined in the Core Services in Jazz for Service Management - WebSphere Application Server profile configuration. This is also the default value proposed when you installed the Dynamic Workload Console, version 8.6 instance. If you maintained this value, then the values in the two instances (8.6 and current) are aligned and no changes are necessary. However, if you used a value different from the default value when you installed the Dynamic Workload Console, version 8.6 instance, then you must necessarily change the default

value proposed during the Dynamic Workload Console, new installation to match the version 8.6 instance. You can verify the value of the server name by checking the name in the place of **server1** in the following path:
<TWS_home_directory>/eWAS/profiles/twaprofile/config/cells/DefaultNode/nodes/DefaultNode/servers/server1.

6. Copy the following files created on the old workstation to the new workstation in the <BACKUP_DIR> backup directory that you want to use for the upgrade process:
 - upgradeData.zip created in step 2 on page 288.
 - HostProperties_file created in step 3 on page 289.

By default, the backup directory used in the upgrade process is:

On Windows operating systems:

<DWC_INSTALL_DIR>\TDWC\tmp\backup

On UNIX and Linux operating systems:

<DWC_INSTALL_DIR>/TDWC/tmp/backup

7. Ensure that no Dynamic Workload Console user interface is active and that the WebSphere Application Server is up and running on the workstation where the back-level Dynamic Workload Console is installed.
8. From the new workstation perform the following steps:

On Windows operating systems:

Map the network drive <DWC_BACKLEV_INSTALL_DIR> of the old workstation where the back-level Dynamic Workload Console is installed.

On UNIX and Linux operating systems:

Mount in read-write access mode the remote file system <DWC_BACKLEV_INSTALL_DIR> where the back-level Dynamic Workload Console is installed. If the mount point name on the new workstation is different from the remote file system name, create a symbolic link between the mount point on the new workstation and the remote file system <DWC_BACKLEV_INSTALL_DIR>;the link name value must be <DWC_BACKLEV_INSTALL_DIR>.

9. Migrate the data from the back-level to the newly-installed Dynamic Workload Console by running the following script from the new workstation:

On Windows operating systems:

```
<DWC_INSTALL_DIR>\TDWC\scripts\tdwclUpgrade.bat
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

On UNIX and Linux operating systems:

```
<DWC_INSTALL_DIR>/TDWC/scripts/tdwclUpgrade.sh
-oldwasuser old_user
-oldwaspassword old_password
-oldtwapath old_twa_path
-newwasuser new_user
-newwaspassword new_password
-newtwapath new_twa_path
[-backuppath backup_path]
-machinechange true
```

where:

-oldwasuser *old_user*

The Tivoli Integrated Portal administrator user ID specified for the back-level Dynamic Workload Console.

-oldwaspassword *old_password*

The Tivoli Integrated Portal administrator user password specified for the back-level Dynamic Workload Console.

-oldtwapath *old_twa_path*

The installation directory where the back-level Dynamic Workload Console is installed.

-newwasuser *new_user*

The Dashboard Application Services Hub administrator user ID. The new user ID must be different from the old one.

-newwaspassword *new_password*

The password of the Dashboard Application Services Hub administrator.

-newtwapath *new_twa_path*

The installation directory where the Dynamic Workload Console must be installed. By default the installation directory is:

On Windows operating systems:

C:\Program Files\IBM\TWAUI

On UNIX and Linux operating systems:

/opt/IBM/TWAUI

-backuppath *backup_path*

The <BACKUP_DIR> backup directory for the upgrade where you already copied the upgradeData.zip and HostProperties_file in step 6 on page 290.

By default, the installation directory is:

On Windows operating systems:

<DWC_INSTALL_DIR>\TDWC\tmp\backup

On UNIX and Linux operating systems:

<DWC_INSTALL_DIR>/TDWC/tmp/backup

This directory contains the following files:

- The **tdwcUpgrade** script log file, upgrade.log.
- The files containing the following configuration data exported from the back-level Dynamic Workload Console:
 - The embedded WebSphere Application Server profile.
 - The embedded WebSphere Application Server profile registry.
 - The port settings in HostProperties_file file.
 - The Tivoli Integrated Portal settings.
 - The Dynamic Workload Console settings.

This data is then imported into the newly-installed Dynamic Workload Console.

- A compressed file named backup.zip that contains a saved copy of the configuration data of the newly-installed Dynamic Workload Console. This file is used to roll back to the original configuration if the

migration script fails while importing the configuration data from the back-level Dynamic Workload Console.

Note: Because the backup.zip file is overwritten every time you run the migration script, it might be useful to save a copy of the first backup.zip file containing the original configuration.

-machinechange true

You must specify the true value to upgrade the Dynamic Workload Console on the new workstation.

Note: The script replaces any customized data in the new Dynamic Workload Console instance, with the data exported from the old Dynamic Workload Console instance. A result of Completed indicates that the script ran successfully and that the data was correctly imported into the newly-installed Dynamic Workload Console.

If the script fails to import the configuration data into the newly-installed Dynamic Workload Console, a rollback is automatically performed and the original configuration is restored. To double-check that the rollback ran correctly, ensure that you can access the newly-installed Dynamic Workload Console user interface with the user ID and password specified during the installation.

10. Check that the data was correctly migrated from the old Dynamic Workload Console to the newly-installed Dynamic Workload Console.
11. Perform this step only if the Dynamic Workload Console ports of the instance installed in the new workstation are different from these of the instance installed on the old workstation and you want to have the same values.
The port numbers used by the two instances of the Dynamic Workload Console might be different and they are not automatically migrated by the **tdwcUpgrade** script. If the Dynamic Workload Console ports of the instance installed on the new workstation are different from the Dynamic Workload Console ports of the instance installed on the old machine and you want to have the same values, run the following steps to migrate the port numbers of the back level instance on the old machine to the newly-installed instance on a new machine:

On Windows operating systems:

```
changeHostProperties <BACKUP_DIR>\HostProperties_file
```

On UNIX and Linux operating systems:

```
changeHostProperties <BACKUP_DIR>/HostProperties_file
```

For more information about this command, see Application server - using the utilities that change the properties.

12. Optionally uninstall the back-level Dynamic Workload Console in the old machine.

Upgrading Dynamic Workload Console V8.6 installed with one or more components in the same directory

Due to Dynamic Workload Console installation infrastructure changes described in “Upgrading overview” on page 281, to upgrade Dynamic Workload Console V8.6 installed with one or more components in the same directory <TWS_INST_DIR> with the previous versions of the Dynamic Workload Console, you must follow the procedures listed in Table 47 on page 293.

Table 47. Upgrade deployment model for Dynamic Workload Console multiple instances in the same directory

Dynamic Workload Console multiple instance installed in the <TWS_INST_DIR> directory contains:	Procedure to run:
Fault-tolerant agent V8.6.0 and related fix packs Dynamic Workload Console V8.6.0 and related fix packs	"Upgrading the fault-tolerant agent and the Dynamic Workload Console"
Fault-tolerant agent V8.6.0 and related fix packs Dynamic Workload Console V8.6.0 and related fix packs z/OS connector V8.6.0 and related fix packs	"Upgrading the fault-tolerant agent and the Dynamic Workload Console."
dynamic domain manager V8.6.0 and related fix packs Dynamic Workload Console V8.6.0 and related fix packs	"Upgrading the dynamic domain manager and the Dynamic Workload Console" on page 294.
dynamic domain manager V8.6.0 and related fix packs Dynamic Workload Console V8.6.0 and related fix packs z/OS connector V8.6.0 and related fix packs	"Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector" on page 217.
Dynamic Workload Console V8.6.0 and related fix packs z/OS connector V8.6.0 and related fix packs	"Procedure to upgrade the Dynamic Workload Console and the z/OS connector" on page 219

Upgrading the fault-tolerant agent and the Dynamic Workload Console

How to upgrade the fault-tolerant agent and the Dynamic Workload Console installed in the same directory.

About this task

If you have a multiple components instance that contains a Dynamic Workload Console and a fault-tolerant agent installed in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory, uninstall the old Dynamic Workload Console installed in the <TWS_INST_DIR> directory, and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

If you try to upgrade the fault-tolerant agent first, the **twsinst** script stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
3. Upgrade the fault-tolerant agent by using the **twsinst** script, as described in Upgrading agents and domain managers.

Upgrading the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector

How to upgrade the fault-tolerant agent, the Dynamic Workload Console, and the z/OS connector installed in the same directory.

About this task

If you installed a multiple components instance that contains a Dynamic Workload Console, a z/OS connector, and a fault-tolerant agent in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory and then upgrade the fault-tolerant agent in the <TWS_INST_DIR> directory.

The Dynamic Workload Console upgrade process migrates also the z/OS connector configuration properties.

If you try to upgrade the fault-tolerant agent first, the **twsinst** script stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
4. Upgrade the fault-tolerant agent by using the **twsinst** script as described in Upgrading agents and domain managers in *IBM Workload Scheduler: Planning and Installation*.

Upgrading the dynamic domain manager and the Dynamic Workload Console

How to upgrade the dynamic domain manager and the Dynamic Workload Console installed in the same directory.

About this task

If you installed a multiple components instance that contains a Dynamic Workload Console and a dynamic domain manager installed in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory, uninstall the old Dynamic Workload Console installed in the <TWS_INST_DIR> directory, and then upgrade the dynamic domain manager in the <TWS_INST_DIR> directory.

If you try to upgrade the dynamic domain manager first, the installation process stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.
3. Upgrade the dynamic domain manager as described in Upgrading a dynamic domain manager instance or its backup .

Upgrading the dynamic domain manager, the Dynamic Workload Console and the z/OS connector

How to upgrade the dynamic domain manager, the Dynamic Workload Console and the z/OS connector installed in the same directory.

About this task

If you installed a multiple component instance that contains a Dynamic Workload Console, a z/OS connector, and a dynamic domain manager in the directory <TWS_INST_DIR>, you must first upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory and then upgrade the dynamic domain manager in the <TWS_INST_DIR> directory.

The Dynamic Workload Console upgrade process migrates also the z/OS connector configuration properties.

If you try to upgrade the dynamic domain manager first, the installation process stops at the beginning and issues an error message that tells you the correct order in which to upgrade the components.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the Dynamic Workload Console previous version uninstallation process.

3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the z/OS connector previous version uninstallation process.
4. Upgrade the dynamic domain manager, as described in *Upgrading a dynamic domain manager instance or its backup in IBM Workload Scheduler: Planning and Installation*.

Procedure to upgrade the Dynamic Workload Console and the z/OS connector

About this task

If you installed a multiple components instance that contains a Dynamic Workload Console and a z/OS connector in the directory <TWS_INST_DIR>, you must upgrade the Dynamic Workload Console in the new <DWC_NEW_INST_DIR> directory.

The Dynamic Workload Console upgrade process also migrates the z/OS connector configuration properties.

To upgrade the multiple components instance in the correct order, perform the following steps:

1. Upgrade the Dynamic Workload Console instance in the new directory <DWC_NEW_INST_DIR>, as described in “Upgrading a Dynamic Workload Console V8.6 single instance” on page 282.
2. Manually uninstall the old Dynamic Workload Console in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the old versions.
3. Manually uninstall the old z/OS connector instance in the directory <TWS_INST_DIR>, by using the uninstallation process provided by the old versions.

Upgrading Dynamic Workload Console V9.1 or later

Before you begin

Ensure that you have upgraded the following software to the latest version, if they apply to your environment. For details about the supported versions, see the Detailed System Requirements for Dynamic Workload Console at the following link <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045182>.

- Jazz for Service Management extension for WebSphere
- Dashboard Application Services Hub

About this task

To upgrade Dynamic Workload Console V9.1 or later to the current version, complete the following procedure.

Procedure

1. Copy the new Dynamic Workload Console image in the <DWC_INSTALL_MEDIA> directory on the system where the back-level Dynamic Workload Console is installed.
2. Start Installation Manager.

3. Create a new Repository Connection pointing to the <DWC_INSTALL_MEDIA> directory:
 - a. Open the Installation Manager wizard.
 - b. Select File -> Preferences ->Repositories. The Repositories page is displayed.
 - c. Select Add Repositories to display the Add Repository page.
 - d. Click Browse and point to the <DWC_INSTALL_MEDIA> directory.
 - e. Select repository.config and click OK to save the new repository.
4. On the Installation Manager main screen select Update.
5. Select Dynamic Workload Console and then Dynamic Workload Console 9.x.
6. Fill in the values (as for the installation process).

Chapter 17. Uninstalling

Uninstalling the Dynamic Workload Console.

How to uninstall the Dynamic Workload Console.

Uninstalling using the Installation Manager wizard

Steps to uninstall using the Installation Manager wizard.

By using the Installation Manager wizard, you can uninstall the installed packages from a single package group, or you can uninstall all installed packages from every package group.

To uninstall a Dynamic Workload Console, perform the following steps:

1. Start the Installation Manager.
2. On the Installation Manager Start page wizard, click **Uninstall**.
3. In the Uninstallation Packages wizard panel, select the Dynamic Workload Console package that you want to uninstall.

Note: If you want to uninstall every package from every package group on your workstation, click **Select all**.

4. Click **Next** to continue.
5. Supply the required fields of the following panel:
 - “WebSphere Application Server profile configuration” on page 263.
6. On the Summary page, review the packages that you selected to uninstall. Click **Back** if you want to make some changes. If you are satisfied with your choices, click **Uninstall**. A progress indicator bar shows the percentage of the uninstallation completed.
7. When the uninstallation process is complete, the Complete page opens and confirms success of the uninstallation process.

Uninstalling in silent mode

Steps to carry out a silent uninstallation.

To perform a silent uninstallation by using a response file template listed in Table 48 on page 300, perform the following steps:

1. Copy the relevant response file to a local directory `<local_dir>` and edit the file to meet the needs of your environment.
2. Save the file with your changes.
3. Open a command-line utility.
4. Go to the Installation Managertools directory.

The default tools directory is:

On Windows operating systems

C:\Program Files\IBM\Installation Manager\eclipse\tools

On UNIX and Linux operating systems

/opt/IBM/InstallationManager/eclipse/tools

5. Run the following command:

On Windows operating systems

```
imcl.exe input <local_dir>\response_file.xml  
-log <local_dir>\log_file.xml  
-acceptLicense
```

On UNIX and Linux operating systems

```
./imcl input /<local_dir>/response_file.xml  
-log /<local_dir>/log_file.xml  
-acceptLicense
```

where

- The response_file.xml is the name of the response file to be used for uninstallation.
- The log_file is the name of the log file that records the result of the silent uninstall execution. For more information about Installation Manager silent log files, see Installation Manager wizard, silent installation and uninstallation log files.

Note: For more information about the Installation Manager silent uninstall command, see http://pic.dhe.ibm.com/infocenter/install/v1r6/index.jsp?topic=/com.ibm.silentinstall12.doc/topics/r_silent_prefs.html.

Table 48 lists the response files to be used for the uninstallation process by platform:

Table 48. Uninstallation response files

Type of installation	Response file to use
Uninstalling on Windows operating systems	
Dynamic Workload Console	TWS92_UNINST_DWC.xml
Uninstalling on UNIX and Linux operating systems	
Dynamic Workload Console	TWS92_UNINST_DWC.xml

Chapter 18. Troubleshooting the installation, upgrade, and uninstallation

Troubleshooting the installation, upgrade, and uninstallation the Dynamic Workload Console.

How to troubleshoot the installation, upgrade, and uninstallation of the Dynamic Workload Console.

Note: To manually uninstall or recover from a failed installation, see the section “Manually uninstall the Dynamic Workload Console and the zConnector on Windows systems” or “Manually uninstall the Dynamic Workload Console and the zConnector on UNIX systems” on page 302

Installation and uninstallation log and trace files

For information about installation log files, see “Installation log files” on page 270.

Manually uninstall the Dynamic Workload Console and the zConnector on Windows systems

To uninstall the Dynamic Workload Console manually and the zConnector on Windows systems.

Run the following steps to manually remove an instance of the Dynamic Workload Console and the zConnector:

1. Start the JazzSM WebSphere Application Server profile on the system where the Dynamic Workload Console and the zConnector are installed.

You can skip this step if the JazzSM WebSphere Application Server profile is already started.

1. In a system prompt, go to the Dynamic Workload Console installation path, for example C:\Program Files\IBM\TWAUI.
2. Go to the `wastools` subdirectory.
3. Run the following command to start the JazzSM WebSphere Application Server profile:

```
startWas.bat -direct
```

2. Uninstall the zConnector package.

1. In a system prompt, go to the Dynamic Workload Console installation path, for example C:\Program Files\IBM\JazzSM\profile.
2. Access the `bin` subdirectory.
3. Run the following command to uninstall the zConnector:

```
wsadmin.bat -conntype NONE -c "$AdminApp uninstall ZConnector"
```

3. Uninstall the zConnector resource adapter.

1. Go to the `wastools` subdirectory under the Dynamic Workload Console installation path.
2. Run the following command to uninstall the zConnector resource adapter:

```
uninstallResourceAdapter.bat -user <your DWC username>
-password <your DWC password>
```

3. Go to the `%JazzSM_profile_dir%\installedApps\%cell_name%` directory, for example: `C:\Program Files\IBM\JazzSM\profile\installedApps\JazzSMNode01Cell\`, and ensure that the `ZConnector.ear` directory is deleted. If it still exists, delete it manually.

4. Uninstall the Dynamic Workload Console package

1. Go to the `bin` subdirectory under the **JazzSM** profile directory, for example, `C:\Program Files\IBM\JazzSM\profile`.
2. Run the following command to uninstall the Dynamic Workload Console package:

```
wsadmin.bat -connType NONE -lang jython -f
"%tdwc_install_dir\TDWC\scripts\install_webui.py"
-war "TWSWebUI.war" -contenturi TWSWebUI.war
-contextroot \ibm\TWSWebUI -serverName %server_name%
-operation delete
```

3. Go to the `C:\%JazzSM_profile_dir%\installedApps\%cell_name%\isc.ear` directory and ensure that the `TWSWebUI.war` directory is deleted. If it still exists, delete it manually.

5. Uninstall the dynamic workload broker package

1. Run the following command to uninstall the Dynamic Workload Console package:

```
wsadmin.bat -connType NONE -lang jython -f
"%tdwc_install_dir\TDWC\scripts\install_webui.py"
-war "WebUI.war" -contenturi WebUI.war -contextroot
\ibm\TDWB -serverName %server_name%
-operation delete
```

2. Go to the `C:\%JazzSM_profile_dir%\installedApps\%cell_name%\isc.ear` directory and ensure that the `WebUI.war` directory is deleted. If it still exists, delete it manually.

6. Uninstall the dynamic workload broker package

Go to the `C:\Windows\TWA` directory and ensure that the files named `twainstance.twa.properties` and `twainstance.twa.properties.ext` apply to the Dynamic Workload Console instance being deleted. For example, you might check that the `TDWC_basePath` key is the same as the `%tdwc_install_dir%` directory. If they do, delete them manually.

7. Delete the Dynamic Workload Console installation directory

Remove manually the `%tdwc_install_dir%` directory, for example `C:\Program Files\IBM\TWAUI`.

Manually uninstall the Dynamic Workload Console and the zConnector on UNIX systems

Steps to manually uninstall the Dynamic Workload Console and the zConnector on UNIX systems.

Run the following steps to manually remove an instance of the Dynamic Workload Console and the zConnector:

1. Start the JazzSM WebSphere Application Server profile on the system where the Dynamic Workload Console and the zConnector are installed.

You can skip this step if the **JazzSM** WebSphere Application Server profile is already started.

1. In a system prompt, go to the Dynamic Workload Console installation path, for example /opt/IBM/TWUI.
2. Go to the wastools subdirectory.
3. Run the following command to start the **JazzSM** WebSphere Application Server profile:
startWas.sh -direct

2. Uninstall the zConnector package.

1. In a system prompt, go to the Dynamic Workload Console installation path, for example /opt/IBM/JazzSM/profile.
2. Access the bin subdirectory.
3. Run the following command to uninstall the zConnector:
wsadmin.sh -connType NONE -c "\\$AdminApp uninstall ZConnector"

3. Uninstall the zConnector resource adapter.

1. Go to the wastools subdirectory under the Dynamic Workload Console installation path.
2. Run the following command to uninstall the zConnector resource adapter:
uninstallResourceAdapter.sh -user <your DWC username>
-password <your DWC password>
3. Go to the \$JazzSM_profile_dir/installedApps/\$cell_name directory, for example /opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell.
4. Verify that the ZConnector.ear directory is deleted. If it still exists, delete it manually.

4. Uninstall the Dynamic Workload Console package.

1. Go to the bin subdirectory under the **JazzSM** profile directory, for example, /opt/IBM/JazzSM/profile/bin.
2. Run the following command to uninstall the Dynamic Workload Console package:

```
wsadmin.sh -connType NONE -lang jython -f "$tdwc_
install_dir/tdwc/scripts/install_webui.py" -war
"$websphere_install_dir/AppServer/systemApps/WebUI.war"
-contenturi TWSWebUI.war -contextroot /ibm/TWSWebUI
-serverName $server_name -operation delete
```

where:

\$tdwc_install_dir

Is the directory on the filesystem where the Dynamic Workload Console is installed, for example, /opt/IBM/TWUI.

\$server_name

Is the name of the Dynamic Workload Console server, for example, **server1**.

3. Go to the \$JazzSM_profile_dir/installedApps/\$cell_name/isc.ear directory.
4. Verify that the TWSWebUI.war directory is deleted. If it still exists, delete it manually.

5. Uninstall the dynamic workload broker package

1. Run the following command to uninstall the Dynamic Workload Console package:

```
wsadmin.sh -connType NONE -lang jython -f
"$tdwc_install_dir/tdwc/scripts/install_webui.py"
-war "$websphere_install_dir/AppServer/systemApps/WebUI.war"
-contenturi WebUI.war -contextroot /ibm/TDWB
-serverName $server_name -operation delete
```

2. Go to the `$JazzSM_profile_dir/installedApps/$cell_name/isc.ear` directory and ensure that the `WebUI.war` directory is deleted. If it still exists, delete it manually.

6. Uninstall the dynamic workload broker package

Go to the `/etc/TWA` directory and ensure that the files named `twainstance.twa.properties` and `twainstance.twa.properties.ext` apply to the Dynamic Workload Console instance being deleted, for example you might check that the `TDWC_basePath` key is the same as the `$tdwc_install_dir` directory. If they do, delete them manually.

7. Delete the Dynamic Workload Console installation directory

Remove manually the `$tdwc_install_dir` directory, for example, `/opt/IBM/TWAUI`.

Troubleshooting scenarios

The troubleshooting scenarios to manage.

Problems with the launchpad

Some problems with the launchpad might be encountered when installing the Dynamic Workload Console.

The following problems might be encountered while using the launchpad to install the Dynamic Workload Console:

- "Warning messages displayed when using the launchpad on Linux systems."
- "Undefined error when using launchpad on Windows operating systems."

Warning messages displayed when using the launchpad on Linux systems

Problem description:

Warning messages might be displayed on the standard output when using the launchpad on Linux systems.

Cause and solution

You can ignore these messages because they do not indicate a malfunction of the launchpad.

Undefined error when using launchpad on Windows operating system

Problem description:

You try to install the Dynamic Workload Console on a Windows operating system using the launchpad and you get an "Undefined" error message. The launchpad does not start.

Cause and solution

Make sure that the path from where you launched the installation does not contain folder names longer than eight characters. If it does, then map the path to the launchpad.exe, and run the launchpad from that new path.

Problems with the interactive installation

Problems that you might encounter while installing the Dynamic Workload Console interactively.

The Dynamic Workload Console installation fails

Problem description:

The installation of the Dynamic Workload Console does not proceed. This occurs regardless of the method you used to install.

Cause and solution

Make sure an active personal firewall is not preventing the installation process from connecting to the network. If it is, allow the connection and then continue with the installation.

Problems with the silent installation

Problems that you might encounter while running the Dynamic Workload Console silent installation.

The silent uninstallation does not work and an error code is returned

Problem description:

If you try to perform a silent uninstall with a response file that does not exist, either because the file name is incorrect or because you specified the wrong directory, an error code is returned and the uninstallation does not run. Nothing is logged in the temporary directory and no messages are issued.

Cause and solution

Ensure that you specify a valid response file name.

Part 5. IBM Workload Scheduler for z/OS Agent

Installing IBM Workload Scheduler for z/OS Agent (also known as agent with z-centric capabilities)

Note: If you are installing on IBM i systems, see Part 6, “IBM Workload Scheduler for z/OS Agent on IBM i systems,” on page 331.

Chapter 19. Installing the IBM Workload Scheduler for z/OS Agent

This chapter describes how to perform a first-time installation of the current version of IBM Workload Scheduler for z/OS Agent (also known as z-centric agents).

You install this agent to run workload from the mainframe to distributed systems with a low cost of ownership.

Using this agent you can run your workload:

Statically

To run existing job types, for example scripts, on a specific IBM Workload Scheduler for z/OS Agent. In this case, you install the IBM Workload Scheduler for z/OS Agent on the distributed systems and connect it to the z/OS system through the IBM Workload Scheduler for z/OS controller.

Statically including job types with advanced options

In this case, you install the IBM Workload Scheduler for z/OS Agent on the distributed systems adding the Java run time, and connect it to the z/OS system through the IBM Workload Scheduler for z/OS controller.

Dynamically

To run existing job types allowing the product to assign them to the workstation that best meets both the hardware and software requirements needed to run them. In this case, you install the IBM Workload Scheduler for z/OS Agent on the distributed systems adding the dynamic capabilities, and connect it to the dynamic domain manager. For a detailed description about how to install a dynamic domain manager for a z/OS controller, see the *IBM Workload Scheduler: Planning and Installation*.

During the installation of the dynamic domain manager for a z/OS controller, you must provide the **master domain manager** and the **IBM Workload Scheduler Netman port** values, even if these values are not used in a z/OS lightweight end-to-end configuration because the fault-tolerant agent is not needed.

Dynamically including job types with advanced options

To run existing job types and job types with advanced options allowing the product to assign them to the workstation that best meets both the hardware and software requirements needed to run them. In this case, you install the IBM Workload Scheduler for z/OS Agent on the distributed systems adding the dynamic capabilities and the Java run time, then connecting it to the dynamic domain manager. For a detailed description about how to install a dynamic domain manager for a z/OS controller, see the *IBM Workload Scheduler: Planning and Installation*.

During the installation of the dynamic domain manager for a z/OS controller, you must provide the **master domain manager** and the **IBM Workload Scheduler Netman port** values, even if these values are not used in a z/OS lightweight end-to-end configuration because the fault-tolerant agent is not needed.

For information how to use it see *Scheduling End-to-end with z-centric Capabilities*.

User authorization requirements

Check the authorization roles before beginning the installation procedure.

Authorization roles for running the `twinst` script

The following table provides the authorization roles required to use the `twinst` method.

Table 49. Required authorization roles for running `twinst`

Activity	Required role
Running the <code>twinst</code> script	Windows operating systems Your login account must be a member of the Windows Administrators group or domain administrators with Act as Part of the Operating System . UNIX and Linux operating systems root access.

Installing using `twinst`

You use the `twinst` script to install IBM Workload Scheduler for z/OS Agent.

Optionally, you can add to the IBM Workload Scheduler for z/OS Agent:

- Dynamic capabilities, to provide your distributed environment with dynamic scheduling capabilities.
- Java run time to run job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins. The run time environment also enables the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

For a complete list of the supported operating systems, see the Detailed System Requirements document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

`twinst`

During the installation process, if you do not specify the installation directory in the command, `twinst` creates files in the following directories for each of the installation steps:

On Windows operating systems:

`%ProgramFiles%\IBM\TWA\TWS`

On UNIX and Linux operating systems:

`/user's_home/TWS`

If you stop and restart the installation, the installation process starts from the installation step where it was stopped.

To install the IBM Workload Scheduler for z/OS Agent and all the supported language packs, perform the following steps:

On Windows operating systems:

1. Insert the DVD for your operating system or download the agent eImage.
2. Log in as administrator on the workstation where you want to install the product.
3. From *DVD_root/TWS/operating_system*, run **twinst** using the synopsis described in the following section.

Note: **twinst** for Windows operating systems is a Visual Basic Script (VBS) that you can run in CScript and WScript mode.

The IBM Workload Scheduler for z/OS user is automatically created. The software is installed by default in the IBM Workload Scheduler for z/OS installation directory. The default value is %ProgramFiles%\IBM\TWA.

On UNIX and Linux operating systems:

1. Insert the DVD for your operating system or download the agent eImage.
2. Create the IBM Workload Scheduler for z/OS user. The software is installed by default in the user's home directory, referred to as */installation_dir/TWS*

User: *TWS_user*

Home: */installation_dir/TWS* (for example: */home/user1/TWS* where *user1* is the name of the IBM Workload Scheduler for z/OS user.)

3. Log in as root on the workstation where you want to install the product.
4. From the *DVD_root/TWS/operating_system* directory, run **twinst** using the synopsis described in the following section.

A successful installation using the **twinst** script issues the return code RC = 0. If the installation fails to understand the cause of the error, see “Analyzing return codes for agent installation, upgrade, restore, and uninstallation” on page 327.

Synopsis

On Windows operating systems:

Show command usage and version

```
twinst -u | -v
```

Install a new instance

```
twinst -new -uname user_name
      -password user_password
      [-addruntime true|false]
      [-displayname agentname]
      [-domain user_domain]
      [-hostname hostname]
      [-inst_dir install_directory]
      [-jimport port_number]
      [-jimportssl true|false]
      [-lang lang_id]
      [-skip_usercheck]
      [-stoponcheckprereq]
      [-tdwport tdwport_number]
      [-tdwhostname hostname]
      [-work_dir working_directory]
```

On UNIX and Linux operating systems:

Show command usage and version

```
twinst -u | -v
```

Install a new instance

```
twinst -new -uname user_name
[-addruntime true|false]
[-displayname agentname]
[-hostname hostname]
[-inst_dir install_directory]
[-jimport port_number]
[-jimportssl true|false]
[-lang lang_id]
[-reset_perm]
[-skip_usercheck]
[-stoponcheckprereq]
[-tdwport tdwport_number]
[-tdwhostname hostname]
[-work_dir working_directory]
```

Parameters

-addruntime *true|false*

Adds the Java runtime to run job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins. Valid values are **true** and **false**. The default is **true**.

If you decided not to install Java runtime, you can still add this feature at a later time, as described in "Part 2. IBM Workload Scheduler -> Chapter 7. Configuring -> Adding a feature" in *Planning and Installation Guide*.

-domain *user_domain*

Windows operating systems only. The domain name of the IBM Workload Scheduler user. The default is the name of the workstation on which you are installing the IBM Workload Scheduler for z/OS Agent.

-displayname *agentname*

The name to be assigned to the IBM Workload Scheduler for z/OS Agent. The default is the host name.

-hostname *hostname*

The fully qualified host name or IP address on which the agent will be contacted by the dynamic workload broker.

-inst_dir *install_directory*

The directory where to install the IBM Workload Scheduler for z/OS Agent. On UNIX and Linux, this path cannot contain blanks. On Windows operating systems, if you specify a path that contains blanks, enclose it in double quotes. If you do not manually specify a path, the path is set to the default home directory. On UNIX and Linux, the path is set to the *user_name* home directory, and on Windows operating systems it is set to %ProgramFiles%\IBM\TWA.

-jimport *port_number*

The port used by the IBM Workload Scheduler for z/OS controller or the dynamic workload broker to connect to the IBM Workload Scheduler for z/OS Agent. The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl *true|false*

The port used by the IBM Workload Scheduler for z/OS controller, or by the dynamic workload broker to connect to the IBM Workload Scheduler

for z/OS Agent. This number is registered in the `ita.ini` file, located in `ITA\cpa\ita` on Windows operating systems and `ITA/cpa/ita` on UNIX and Linux.

For communication using SSL or HTTPS

Set `jmportssl = true`. To communicate with the dynamic workload broker, it is recommended that you set the value to `true`. In this case, the port specified in `jmport` communicates in HTTPS. If you specify `true`, ensure that you also configure the HTTPS communication on the z/OS master.

For communication without using SSL, or through HTTP

Set `jmportssl = false`. In this case the port specified in `jmport` communicates in HTTP.

The default value is `true`.

To increase the performance of the IBM Workload Scheduler for z/OS server, it is recommended that you set this value to `false`.

-lang *lang_id*

The language in which the `twinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: This is the language in which the installation log is recorded, and not the language of the installed engine instance. `twinst` installs all languages as default.

-new A fresh installation of the agent. Installs an agent and all supported language packs.

-password *user_password*

Windows operating systems only. The password of the user for which you are installing IBM Workload Scheduler for z/OS Agent.

-reset_perm

UNIX and Linux only. Reset the permissions of the `libatrc` library.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX and Linux operating systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the `su` command. On Windows operating systems if you specify this parameter, the program does not create the user you specified in the `-uname username` parameter. If you specify this parameter you must create the user manually before running the script.

-stoponcheckprereq

Stop the installation whenever a problem occurs during the prerequisite check. For more information about the prerequisites, see <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.

-tdwbhostname *hostname*

The fully qualified host name of the dynamic domain manager of backup dynamic domain manager used to connect to the IBM Workload Scheduler for z/OS Agent. It is used together with the `-tdwbport tdwbport_number` parameter. It adds the capability to run dynamic workload to the IBM

Workload Scheduler for z/OS Agent. If not specified, the default value is **localhost**. This value is registered in the ResourceAdvisorUrl property in the JobManager.ini file.

-tdwbport *tdwbport_number*

The dynamic domain manager of backup dynamic domain manager HTTP or HTTPS port number used to connect to the IBM Workload Scheduler for z/OS Agent. It is used together with the -tdwbhostname *host_name* parameter to add the capability to run dynamic workload to the IBM Workload Scheduler for z/OS Agent. This number is registered in the ResourceAdvisorUrl property in the JobManager.ini file. The default value is **0**, meaning that the capability to run dynamic workload to the agent is not added. The valid range is from 0 to 65535.

-u Displays command usage information and exits.

-uname *user_name*

The name of the user for which the IBM Workload Scheduler for z/OS Agent is installed. This user name is not to be confused with the user performing the installation logged on as **root** on UNIX and Linux and as **administrator** on Windows operating systems.

On UNIX and Linux, this user account must be created manually before running the installation. Create a user with a home directory. By default, IBM Workload Scheduler for z/OS Agent is installed in the home directory of the specified user.

-work_dir *working_directory*

The temporary directory used for the IBM Workload Scheduler installation process files deployment.

On Windows operating systems

If you specify a path that contains blanks, enclose it in double quotes. If you do not manually specify a path, the path is set to %temp%\TWA\tws93, where %temp% is the temporary directory of the operating system.

On UNIX and Linux operating systems

The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/TWA/tws93.

-v Displays the command version and exits.

Examples

This example describes how to install the IBM Workload Scheduler for z/OS Agent for the user TWS_user and accept the default value to add the runtime environment for Java. The runtime environment is used to run jobs supplied with the product or implemented through the custom plug-ins, it also enables the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

On Windows operating systems:

```
twsinst -new
-uname TWS_user
-password qaz12qaz
-jmportssl false
-jmport 31114
-inst_dir "c:\Program Files\IBM\TWA\TWS_user"
```

On UNIX and Linux operating systems:

```
./twinst -new
-uname TWS_user
-jmportssl false
-jmport 31114
-inst_dir /home/TWS_user/TWA
```

The twinst log files

About this task

The **twinst** log file name is:

On Windows operating systems:

```
<TWS_INST_DIR>\logs\
twinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is C:\Program Files\IBM\TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

On UNIX and Linux operating systems:

```
<TWS-INST-DIR>/logs/
twinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is /opt/IBM/TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

Creating a Docker image to run IBM Workload Scheduler for z/OS Agents

Quickly create a Docker image to run IBM Workload Scheduler for z/OS Agents.

You can run IBM Workload Scheduler for z/OS Agents in a Docker container that you use to run jobs remotely, for example to call REST APIs or database stored procedures, or to run jobs within the container itself.

To create a Docker container, you are provided with step-by-step instructions and the latest versions of the required samples on GitHub [here](#). Follow the instructions to create a Docker container to run jobs remotely, or use it as base image to add the applications to be run with the agent to other images, or customize the samples to best meet your requirements.

Enabling dynamic capabilities after installation or upgrade

This section describes the procedure that you must perform to enable dynamic scheduling capabilities after you installed or upgraded the IBM Workload Scheduler for z/OS Agent, without enabling them:

1. Update the `JobManager.ini` configuration file located in:

Windows operating systems:

`twshome\TWS\ITA\cpa\config\JobManager.ini`

UNIX and Linux operating systems:

`twshome/TWS/ITA/cpa/config/JobManager.ini`

by assigning to the `tdwb_hostname` and `mdm_httpsport` variables contained in the `ResourceAdvisorUrl` property, the following values:

tdwb_hostname

The fully qualified host name of the workload broker server.

mdm_httpsport

The value that the **httpsPort** has on the master domain manager, as shown by the **showHostProperties** wastool. The default is 31116, which is the dynamic workload broker port number. The port is currently set to zero because at installation time you specified that you would not use the dynamic workload broker.

The **ResourceAdvisorUrl** property has the following syntax:

```
ResourceAdvisorUrl = https://tdwb_hostname:mdm_httpsport  
/JobManagerRESTWeb/JobScheduler/resource
```

2. Start the IBM Workload Scheduler for z/OS Agent by running the following command from *TWS_home*:

Windows operating systems:

`StartUpLwa.cmd`

UNIX and Linux operating systems:

`StartUpLwa`

Installing the additional plug-ins by using the IBM Workload Scheduler for additional plug-ins

About this task

The IBM Workload Scheduler for additional plug-ins is an installation process that you can use to install the additional plug-ins that you have developed to resolve your particular needs. This installer is contained in the IBM Workload Scheduler eImages.

For detailed information, see the section about installing additional plug-ins by using the IBM Workload Scheduler for additional plug-ins in *IBM Workload Scheduler: Planning and Installation*.

Chapter 20. Upgrading the IBM Workload Scheduler for z/OS Agent

Upgrading IBM Workload Scheduler for z/OS Agent (z-centric) from older versions to the current version.

Coexistence with previous versions

The current version of the IBM Workload Scheduler for z/OS Agent (z-centric) can be installed on any workstation containing a prior version, provided that the *TWS_user*, **JobManager** port, and installation path are different from those of the previous versions.

User authorization requirements

Check the authorization roles before beginning the upgrade procedure. For detailed information, see “User authorization requirements” on page 310.

Upgrading notes

Before upgrading the IBM Workload Scheduler for z/OS Agent, ensure that there are no jobs running on the agent.

If you are upgrading IBM Workload Scheduler for z/OS Agent from an installation where you did not install the dynamic capabilities, you cannot add them during the upgrade process. To add them, perform the procedure described in the following section:

- “Enabling dynamic capabilities after installation or upgrade” on page 316

When the upgrade procedure is successful, it is not possible to roll back to the previous version. Rollback is possible only for upgrades that fail.

Upgrading using twsinst

Use **twsinst** to upgrade IBM Workload Scheduler for z/OS Agent by satisfying the following objectives:

Save time, disk space, and RAM when upgrading the product

It saves disk space and RAM because it is not Java-based.

Use a very simple command

It consists of a single line command.

Manage both UNIX and Windows operating system workstations

It runs both on UNIX and Windows agents.

For a list of the supported operating systems and requirements, refer to <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

Upgrading process

According to your operating system, to upgrade the IBM Workload Scheduler for z/OS Agent with **twsinst** perform the following steps:

Windows operating systems

1. Insert the DVD related to your operating system.
2. Log in as administrator on the workstation where you want to upgrade the agent.
3. From the *DVD_root/TWS/operating_system* directory of the DVD, run the **twinst** script using the synopsis described in this section.

Note: **twinst** for Windows is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twinst -update -uname username
-acceptlicense yes
```

UNIX and Linux operating systems

1. Insert the installation DVD related to your operating system.
2. From *DVD_root/TWS/operating_system*, run the **twinst** script using the synopsis described in this section.

A successful upgrade using the **twinst** script issues the return code RC = 0. If the installation fails to understand the cause of the error, see “Analyzing return codes for agent installation, upgrade, restore, and uninstallation” on page 327.

Synopsis:

On Windows operating systems:

Show command usage and version

```
twinst -u | -v
```

Upgrade an instance

```
twinst -update -uname user_name
[-addjruntime true]
[-backup_dir backup_directory]
[-domain user_domain]
[-inst_dir install_directory]
[-lang lang_id]
[-nobackup]
[-recovInstReg true]
[-skip_usercheck]
[-wait minutes]
[-work_dir working_directory]
```

UNIX and Linux operating systems:

Show command usage and version

```
twinst -u | -v
```

Upgrade an instance

```
twinst -update -uname user_name
[-addjruntime true]
[-backup_dir backup_directory]
[-inst_dir install_directory]
[-lang lang-id]
[-nobackup]
[-recovInstReg true]
[-reset_perm]
[-skip_usercheck]
[-wait minutes]
[-work_dir working_directory]
```

-addjruntime true

Adds the Java run time to run job types with advanced options, both those

supplied with the product and the additional types implemented through the custom plug-ins. With the `-update` option, the only valid value for this parameter is **true**.

If you decided not to install Java run time when upgrading, you can still add this feature at a later time. For details about how to add a feature, see the *IBM Workload Scheduler: Planning and installation* manual.

-backup_dir *backup_directory*

An alternative directory (which must be created manually) as the destination for the backup copy of a previous version.

If you do not specify this option when running an upgrade, the following default value is used:

```
$BACKUP_DIR = $INST_DIR_backup_$TWS_USER
```

where:

- `$INST_DIR` is the installation path (the user home directory on UNIX and Linux).
- `$TWS_USER` is the user name.

For example:

```
$INST_DIR=/opt/TWS/TWS86
$TWS_USER=user86
$BACKUP_DIR=/opt/TWS/TWS86_backup_user86
$BACKUP_SUBDIR=/opt/TWS/TWS86_backup_user86/TWS86
```

In the backup directory you must also create a subdirectory to include as the latest directory of the installation path.

-domain *user_domain*

Windows operating systems only. The domain name of the IBM Workload Scheduler for z/OS Agent user. The default is the name of the workstation on which you are upgrading the agent.

-inst_dir *install_directory*

The directory of the IBM Workload Scheduler for z/OS Agent installation. On UNIX this path cannot contain blanks. On Windows operating systems, if you specify a path that contains blanks, enclose it in a double quotation marks. On any operating system, specify an absolute path. If not specified, On UNIX the path is set to the `/opt/IBM/TWS` directory, on Windows operating systems the path is set to `%ProgramFiles%\IBM\TWA`.

-lang

The language in which the `twsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: The **-lang** option does not relate to the supported language packs. By default, all supported language packs are installed when you install using the `twsinst` script.

-nobackup

The upgrade process does not back up the instance you are upgrading.

-uname *user_name*

The name of the user for which IBM Workload Scheduler for z/OS Agent is being upgraded. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade. This

user name is not to be confused with the user performing the installation logged on as **root** on UNIX and Linux and as **administrator** on Windows operating systems.

-update

Upgrades an existing agent that was installed using **twinst**.

-recovInstReg *true*

To re-create the registry files. Specify it if you have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of IBM Workload Scheduler cannot be found, this can be caused by a corrupt registry file. See Upgrading when there are corrupt registry files in *IBM Workload Scheduler: Planning and Installation*.

-reset_perm

UNIX and Linux only. Reset the permissions of the **libatrc** library.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX and Linux operating systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the **su** command. On Windows operating systems if you specify this parameter, the program does not create the user you specified in the `-uname username` parameter. If you specify this parameter you must create the user manually before running the script.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-work_dir *working_directory*

The temporary directory used by the installation process to store the files to deploy.

On Windows operating systems:

If you specify a path that contains blanks, enclose it in a double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws93`, where `%temp%` is the temporary directory of the operating system.

On UNIX and Linux operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws93`.

Examples

To upgrade the agent installed for the user `TWS_user` in the user home directory that does not have the dynamic scheduling capabilities and the Java runtime to run job types with advanced options, run the following command:

```
./twinst -update -uname TWS_user
```

The twinst log files

About this task

The **twinst** log file name is:

On Windows operating systems:

```
<TWS_INST_DIR>\logs\  
twsinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is C:\Program Files\IBM\TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

On UNIX and Linux operating systems:

```
<TWS-INST-DIR>/logs/  
twsinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is /opt/IBM/TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

Enabling dynamic capabilities after upgrade

To enable dynamic scheduling after you upgraded the IBM Workload Scheduler for z/OS Agent without enabling it, refer to the following section:

- “Enabling dynamic capabilities after installation or upgrade” on page 316

Chapter 21. Uninstalling the IBM Workload Scheduler for z/OS Agent

Uninstalling the agent does not remove files created after the agent was installed, nor files that are open at the time of uninstallation. If you do not need these files, you must remove them manually. If you intend to reinstall and therefore need the files, make a backup before starting the installation process.

User authorization requirements

Check the authorization roles before beginning the uninstallation procedure. See “User authorization requirements” on page 310.

Uninstalling the IBM Workload Scheduler for z/OS Agent using the `twinsinst` script

Follow these steps to uninstall the IBM Workload Scheduler for z/OS Agent using the `twinsinst` script. Depending on the operating system, proceed as follows:

- On Windows operating systems:
 1. Ensure that all IBM Workload Scheduler processes and services are stopped, and that there are no active or pending jobs.
 2. Log on as administrator on the workstation where you want to uninstall the product.
 3. `twinsinst` for Windows operating systems is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, from the `installation_dir/TWS` (for example, `c:\Program Files\IBM\TWA`), run the `twinsinst` script as follows:

```
cscript twinsinst -uninst -uname user_name [-wait minutes]  
[-domain domain_name] [-lang lang_id]  
[-work_dir working_dir]
```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a language other than the locale of the computer, run the `twinsinst` script from the `/installation_dir/TWS` (for example, `/home/user1/TWS`) as follows:

```
cscript twinsinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

- On UNIX and Linux operating systems:
 1. Ensure that all processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see *Administration Guide*.
 2. Log on as root and change your directory to `/installation_dir/TWS` (for example: `/home/user1/TWS` where `user1` is the name of IBM Workload Scheduler user.)
 3. From the TWS directory, run the `twinsinst` script as follows:

```
twinsinst -uninst -uname user_name [-wait minutes]  
[-lang lang_id] [-work_dir working_dir]
```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a

language other than the locale of the computer, run the **twinst** script from the */installation_dir/TWS* (for example, */home/user1/TWS*) as follows:

```
./twinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

-uninst

Uninstalls the agent

-uname *user_name*

The name of the user for which the agent is uninstalled. This user name is not to be confused with the user performing the installation logged on as **root** on UNIX and Linux, and as **administrator** on Windows operating systems.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the uninstallation. If the jobs do not complete during this interval the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-domain *domain_name*

Windows operating systems only. The domain name of the IBM Workload Scheduler user. The default is the name of the workstation on which you are uninstalling the product.

-lang *lang_id*

The language in which the **twinst** messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: The **-lang** option is not to be confused with the IBM Workload Scheduler supported language packs.

-work_dir *working_dir*

The temporary directory used for the installation process files deployment.

On Windows operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to %temp%\TWA\tws93, where %temp% is the temporary directory of the operating system.

On UNIX and Linux operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/TWA/tws93.

The following is an example of a **twinst** script that uninstalls the IBM Workload Scheduler agent, originally installed for user named **TWS_user**:

On Windows operating systems:

```
twinst -uninst -uname TWS_user
```

On UNIX and Linux operating systems:

```
./twinst -uninst -uname TWS_user
```

The twinst log files

About this task

The **twinst** log file name is:

On Windows operating systems:

```
<TWS_INST_DIR>\logs\  
twsinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is C:\Program Files\IBM\TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

On UNIX and Linux operating systems:

```
<TWS-INST-DIR>/logs/  
twsinst_<operating_system>_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is /opt/IBM/TWA_<TWS_user>.

<operating_system>

The operating system.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

Uninstalling the additional plug-ins using the IBM Workload Scheduler for additional plug-ins

You can uninstall the additional plug-ins using either the wizard or the silent method.

About this task

When you uninstall the additional plug-ins, you can uninstall one or more plug-ins simultaneously.

To uninstall an additional plug-in, you can use any of the following procedures:

Wizard

For details, see Uninstalling by using the wizard in IBM Workload Scheduler Planning and Installation.

Silent For details, see Uninstalling by using the silent uninstallation in IBM Workload Scheduler Planning and Installation.

Note: You can uninstall only additional plug-ins installed by using IBM Workload Scheduler for additional plug-ins.

Chapter 22. Troubleshooting and maintaining installation, upgrade, and uninstallation

This chapter describes how to troubleshoot and maintain the installation, the upgrade, and the uninstallation of the agent.

Analyzing return codes for agent installation, upgrade, restore, and uninstallation

Check how your operation completed by analyzing the return codes that are issued by twsinst.

Return codes that you can receive when you are installing, upgrading, restoring, or uninstalling agents. To analyze them and take corrective actions, run the following steps:

On Windows operating systems

1. Display the operation completion return code, by using the following command:
`echo %ERRORLEVEL%`
2. Analyze the following table to verify how the operation completed:

Table 50. Windows operating system agent return codes

Error Code	Description	User action
0	Success: The operation completed successfully without any warnings or errors.	None.
1	Generic failure	Check the messages that are displayed on the screen by the script. Correct the error and rerun the operation. If the error persists, search the IBM Support database for a solution at http://www.ibm.com/software/sysmgmt/products/support .
2	The installation cannot create the IBM Workload Scheduler user or assign the correct permission to it.	Verify the operating system policies and configuration. Verify the input values. If necessary, create the user manually before you run the installation.
3	The password is not correct or the installation cannot verify it.	Verify the operating system policies and configuration. Verify the input values.
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	Check the product system requirements at the following link: http://www.ibm.com/support/docview.wss?rs=672&tuid=swg27023736 .

Table 50. Windows operating system agent return codes (continued)

Error Code	Description	User action
6	The IBM Workload Scheduler registry is corrupted.	Use the <code>recovInstReg</code> option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the <code>localopts</code> , the <code>globalopts</code> , the <code>ita.ini</code> , and the <code>JobManager.ini</code> files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

On UNIX and Linux operating systems:

1. Display the installation completion return code, by using the following command:

```
echo $?
```
2. Analyze the following table to verify how the installation completed:

Table 51. UNIX or Linux operating system agent return codes

Error Code	Description	User action
0	Success: The installation completed successfully without any warnings or errors.	None.
1	Generic failure.	Check the messages that are displayed on the video by the script. Correct the error and rerun the operation. If the error persists, search the IBM Support database for a solution at http://www.ibm.com/software/sysmgmt/products/support .
2	The installation did not find the IBM Workload Scheduler user or its home directory. The IBM Workload Scheduler user that you specified either does not exist or does not have an associated home directory.	Verify the operating system definition of the IBM Workload Scheduler user.
3	Not applicable	
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.

Table 51. UNIX or Linux operating system agent return codes (continued)

Error Code	Description	User action
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	Check the product system requirements at the following link: http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736 .
6	The IBM Workload Scheduler registry is corrupted.	Use the <code>recovInstReg</code> option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the <code>localopts</code> , the <code>globalopts</code> , the <code>ita.ini</code> , and the <code>JobManager.ini</code> files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

twinst needs long time to run if the machine does not have enough temporary space

About this task

Problem:

If the machine does not have enough temporary space, the agent installation performed by using the `twinst` script needs a long time to run, due to concomitant use of the temporary directory by the `twinst` script and by the `check prerequisites` script started by the `twinst` script.

Cause and solution:

You can solve the long time execution problem by manually running the `prereq_checker.sh` script on UNIX and Linux operating systems and `prereq_checker.bat` script on Windows operating systems, that performs the check prerequisites process before running the `twinst`.

You can manually run the check prerequisites script, by performing the following steps:

On Windows operating systems:

1. Log on as Administrator on the machine where you want to install the agent.
2. Go to the `<CD-ROM>\Prerequisites` directory where `<CD-ROM>` is the directory where you mounted the CD-ROM.
3. Run:

Dynamic agent or IBM Workload Scheduler for z/OS Agent

```
prereq_checker.bat "DA1 09010000"  
-p "DA1.inst_dir=<TWS_INST_DIR>,DA1.work_dir=<TEMP_DIR>"
```

where *<TWS_INST_DIR>* is the IBM Workload Scheduler installation directory and *<TEMP_DIR>* is the temporary directory.

Fault tolerant-agent

```
prereq_checker.bat "FTA 09010000"  
-p "FTA.inst_dir=<TWS_INST_DIR>,,FTA.work_dir=<TEMP_DIR>"
```

where *<TWS_INST_DIR>* is the IBM Workload Scheduler installation directory and *<TEMP_DIR>* is the temporary directory.

On UNIX and Linux operating systems:

1. Log on as root on the machine where you want to install the agent.
2. Go to the *<CD-ROM>\Prerequisites* directory where *<CD-ROM>* is the directory where you mounted the CD-ROM.
3. Run:

Dynamic agent or IBM Workload Scheduler for z/OS Agent

```
./prereq_checker.sh "DA1 09010000,TWA 09010000"  
-p "DA1.inst_dir=<TWS_INST_DIR>,DA1.work_dir=<TEMP_DIR>"
```

where *<TWS_INST_DIR>* is the IBM Workload Scheduler installation directory and *<TEMP_DIR>* is the temporary directory.

Fault tolerant-agent

```
./prereq_checker.sh "FTA 09010000,TWA 09010000"  
-p "FTA.inst_dir=<TWS_INST_DIR>,FTA.work_dir=<TEMP_DIR>"
```

where *<TWS_INST_DIR>* is the IBM Workload Scheduler installation directory and *<TEMP_DIR>* is the temporary directory.

Part 6. IBM Workload Scheduler for z/OS Agent on IBM i systems

Installing IBM Workload Scheduler for z/OS Agent on IBM i systems

Chapter 23. Prerequisites

Describes the prerequisites for running the IBM i agent.

About this task

To install and use the IBM i agent you must have a supported version of the IBM i operating system. For a detailed list of supported operating systems, see the Detailed System Requirements document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

Chapter 24. Installing agents on IBM i systems

Learn how to install agents on IBM i systems.

About this task

You install the IBM Workload Scheduler for z/OS agent and dynamic agent on IBM i systems by using the `twinst` installation script.

To install an agent, complete the following steps:

1. Sign on as **QSECOFR** user.
2. Create an IBM i user profile for which the IBM Workload Scheduler agent is installed.

Note: The user profile is not the same as for the user that is performing the installation logged on as **QSECOFR**. Instead the user profile is for the user that you specify in the `-uname username` parameter when running the `twinst` script. For descriptions of the syntax parameters, see “Agent installation parameters on IBM i systems” on page 337. You cannot use an existing IBM i system user profile, an application supplied user profile, or any of the following reserved IBM i user profiles:

- QDBSHR
- QDFTOWN
- QDOC
- QLPAUTO
- QLPINSTALL
- QRJE
- QSECOFR
- QSPL
- QSYS
- QTSTRQS

Attention: Be aware of the following considerations:

- If the user profile is a member of a group, the installation fails. Set the group profile that is associated with the user profile to `*NONE`.
 - If the `username` is longer than 8 characters, after the installation the agent (and the JobManager component) runs under the **QSECOFR** user instead of under the authority of the installation user. To prevent this problem, set the `PASE_USRGRP_LIMITED` environment variable to `N`.
3. On the IBM i system, verify that no library exists with the same name as the user profile supplied for the agent user.
 4. Download the IBM i agent eImage from the Passport Advantage® Online website. For more information about the installation media, see Installation media in *IBM Workload Scheduler: Planning and Installation* or the Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.
 5. To untar or unzip the agent eImage, you can use the `PASE` shell or the `AIXterm` command.


```
[-lang lang_id]
[-tdwport tdwport_number]
[-tdwhostname host_name]
[-work_dir working_dir]
```

For a description of the installation parameters and options that are related to agent on this operating system, see “Agent installation parameters on IBM i systems” in *IBM Workload Scheduler: Planning and Installation*.

Agent installation parameters on IBM i systems

About this task

The parameters set when using the `twinsinst` script to install the dynamic agent on IBM i systems.

-acceptlicense *yes|no*

Specify whether to accept the License Agreement.

-addruntime *true|false*

Adds the Java run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**.

This option is applicable to both fault-tolerant agents and dynamic agents.

If you decided not to install Java run time at installation time, you can still add this feature later. For details about how to add a feature, see IBM Workload Scheduler Planning and Installation.

-company *company_name*

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-displayname

The name to assign to the agent. The default is the host name of this computer.

If the host name starts with a number, **-displayname** parameter must be specified.

-gateway *local|remote|none*

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *local* if the gateway is local to the dynamic agent workstation. Specify *remote* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *none*, no gateway is configured.

-gweifport *gateway_eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is 31132. The valid range is 1 to 65535.

-gwid *gateway_id*

The unique identifier for the gateway. This parameter is required when you specify **-gateway local**. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic

character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (_), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the JobManagerGW.ini file, by setting the JobManagerGWURIs property.

-hostname *host_name*

The fully qualified host name or IP address on which the agent is contacted by the dynamic workload broker. The default is the host name of this computer.

-inst_dir *installation_dir*

The directory of the IBM Workload Scheduler installation. Specify an absolute path. The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the *home/username* directory, where *username* is the value specified in the -uname option.

-jimport *port_number*

The JobManager port number used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl *true|false*

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The port value is the value of the *ssl_port* parameter in the *ita.ini* file if **-jimportssl** is set to true. If set to false, it corresponds to the value of the **tcp_port** parameter in the *ita.ini* file. The *ita.ini* file is located in ITA\cpa\ita on Windows systems and ITA/cpa/ita on UNIX, Linux, and IBM i systems.

Set the value to "true" if **-gateway** is set to local.

For communication using SSL or HTTPS

Set **jimportssl = true**. To communicate with the dynamic workload broker, it is recommended that you set the value to **true**. In this case, the port specified in **jimport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jimportssl = false**. In this case the port specified in **jimport** communicates in HTTP.

-lang *lang_id*

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 52. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en

Table 52. Valid values for *-lang* and *LANG* parameter (continued)

Language	Value
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. **twinst** installs all languages as default.

-new A fresh installation of the agent. Installs an agent and all supported language packs.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent.

For a detailed list of supported operating systems and product prerequisites, see <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

-tdwbhostname *host_name*

The fully qualified host name of the dynamic workload broker. It is used together with the **-agent** *dynamic* and the **-tdwbport** *tdwbport_number* parameters. If not specified, you cannot run your workload dynamically and this parameter uses the **localhost** default value. This value is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file.

If **-gateway** *local* or *remote* is specified, then this is the host name of the dynamic agent workstation where the gateway resides and to which the agent connects. This information is stored in the `JobManager.ini` file.

-tdwbport *tdwbport_number*

The dynamic workload broker HTTP or HTTPS transport port number. It is used together with the **-agent** *dynamic* and the **-tdwbhostname** *host_name* parameters. The valid range is from 0 to 65535. If you specify **0** or do not specify this parameter, you cannot run workload dynamically. Do not specify **0** if the **-agent** value is **dynamic**. This number is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. The default value is **41114**.

If **-gateway** *remote* is specified, then this is the HTTP or HTTPS port number of the dynamic agent workstation where the gateway resides and to which the agent connects. If you are performing a fresh installation, then the value to use is 31114. This information is stored in the `JobManager.ini` file.

- thiscpu** *workstation*
The name of the IBM Workload Scheduler workstation of this installation. The name cannot exceed 16 characters, cannot contain spaces and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.
- u** Displays command usage information and exits.
- uname** *username*
The name of the user for which IBM Workload Scheduler is installed.
- Note:** This user name is not the same as the user performing the installation logged on as **QSECOFR**.
- If *username* is longer than 8 characters, after installation the agent (and the JobManager component) erroneously run under the **QSECOFR** user, instead of under the authority of the installation user. To prevent this, set the `PASE_USRGRP_LIMITED` environment variable to N.
- work_dir** *working_dir*
The temporary directory used for the IBM Workload Scheduler installation process files deployment. The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws93`.
- v** Displays the command version and exits.

Example installation of an agent on IBM i systems

About this task

The following example shows the syntax used when using the **twsinst** script to install a new instance of the agent on IBM i system.

```
./twsinst -new
-uname TWS_user
-acceptlicense yes
-hostname thishostname.mycompany.com
-jmport 31114
-tdwbport 41114
-tdwbhostname mainbroker.mycompany.com
-work_dir "/tmp/TWA/tws93"
```

Chapter 25. Upgrading agents on IBM i systems

How to upgrade agents on IBM i systems.

About this task

You can upgrade the agent on an IBM i system by using the `twsoinst` installation script.

To upgrade an IBM Workload Scheduler agent, perform the following steps:

1. Sign on as **QSECOFR** user.
2. Insert the DVD for the IBM i system or download the agent eImage from the Passport Advantage Online website. For more information about the installation media, see Installation media or the Download Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24039466>.
3. If you downloaded the eImages, to extract the package, use the *PASE* shell or the *AIXterm* command.

Using *PASE* shell:

- a. Open the *PASE* shell.
- b. Run the command **"CALL QP2TERM"**.
- c. Locate the folder where you downloaded the eImages and run the command:

```
"tar xvf TWS93_IBM_I.tar"
```
- d. Exit from the *PASE* shell.

Using *AIXterm* command:

- a. Start the *Xserver* on your desktop.
 - b. On the iSeries machine, open a *QSH shell* and export the display.
 - c. In *QSH shell*, go to the directory */QopenSys* and run the command **"aixterm -sb"**.
 - d. A pop-up window is displayed on your desktop. By Using this pop-up window, extract the file *TWS93_IBM_I.tar*.
4. Open a *QSH shell* and run the `twsoinst` script.
The installation procedure replaces the library to the user profile library list of the dynamic agent user profile and sets this job description as the job description of the dynamic agent user profile. The upgrade process replaces the new version of the agent in the directory where the old agent is installed.

Note: If you do not run the `twsoinst` script from a *QSH shell* the installation fails.

If the operation fails to understand the cause of the error, see "Analyzing return codes for agent installation, upgrade, restore, and uninstallation" on page 327.

Command usage and version

Show command usage and version

```
twsoinst -u | -v
```

Upgrade an instance

```

./twinst -update -uname user_name
-acceptlicense yes|no
[-addjruntime true]
[-create_link]
[-hostname host_name]
[-inst_dir install_dir]
[-jimport port_number]
[-jimportssl boolean]
[-lang lang-id]
[-reset_perm]
[-recovInstReg true]
[-skip_usercheck]
[-tdwbhostname host_name]
[-tdwbport port_number]
[-wait minutes]
[-work_dir working_dir]

```

For a description of the installation parameters and options that are related to agent on this operating system, see “Agent upgrade parameters on IBM i systems.”

Agent upgrade parameters on IBM i systems

About this task

The parameters set when using the `twinst` script to upgrade a dynamic agent on IBM i systems.

-acceptlicense *yes|no*

Specify whether to accept the License Agreement.

-addjruntime *true*

Adds the Java run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

By default, if the Java run time was already installed on the agent, it will be upgraded to the new version.

If the Java run time was not installed on the agent, it will not be installed during the upgrade, unless you specify `-addjruntime true`.

If you decided not to install Java run time when you upgrade, you can still add this feature later. For details about how to add a feature, see *IBM Workload Scheduler for z/OS: Planning and installation*.

-create_link

Create the **symlink** between `/usr/bin/at` and `<install_dir>/TWS/bin/at`. See Symbolic link options for more information.

-displayname

The name to assign to the agent. The default is the host name of this computer.

-inst_dir *installation_dir*

The directory of the IBM Workload Scheduler installation.

Note: The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the `user_home\user_name` directory.

-jimport *port_number*

The JobManager port number used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The default value is 31114. The valid range is from 1 to 65535.

-jimportssl *true|false*

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. This number is registered in the `ita.ini` file located in the `ITA/cpa/ita` directory.

For communication using SSL or HTTPS

Set `jimportssl = true`. To communicate with the dynamic workload broker, it is recommended that you set the value to `true`. If the value is set to `true`, the port specified in `jimport` communicates in HTTPS.

For communication without using SSL, or through HTTP

Set `jimportssl = false`. If the value is set to `false`, the port specified in `jimport` communicates in HTTP.

-lang *lang_id*

The language in which the `twinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: This is the language in which the installation log is recorded, and not the language of the installed engine instance. The `twinst` script installs all languages by default.

-recovInstReg *true*

To re-create the registry files. Specify it if you have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of IBM Workload Scheduler cannot be found, this can be caused by a corrupt registry file. See *Upgrading when there are corrupt registry files*.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before upgrading the agent.

For a detailed list of supported operating systems and product prerequisites, see <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>.

-tdwbhostname *host_name*

The dynamic workload broker fully qualified host name. It is used together with the `-tdwbport` *tdwbport_number* parameter. It adds and starts the capabilities to run workload dynamically to IBM Workload Scheduler. If not specified you cannot run your workload dynamically and this parameter assumes the `localhost` default value. This value is registered in the `ResourceAdvisorUrl` property in the `JobManager.ini` file.

-tdwbport *tdwbport_number*

The dynamic workload broker HTTP or HTTPS port number used to add dynamic scheduling capabilities to your distributed or end-to-end

environment. It is used together with the **-tdwbhostname** *host_name* parameter. This number is registered in the ResourceAdvisorUrl property in the JobManager.ini file. The default value is 0, however, if you leave the value as 0, you cannot run your workload dynamically. Specify a nonzero value to add dynamic capability. The valid range is 0 to 65535.

-uname *user_name*

The name of the user for which IBM Workload Scheduler is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade.

Note: This user name is not the same as the user performing the installation logged on as QSECOFR.

-update

Upgrades an existing agent that was installed using **twsinst**.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or -1 for the product to wait indefinitely. The default is 60 minutes.

-work_dir *working_dir*

The temporary directory used for the IBM Workload Scheduler installation process files deployment. The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/TWA/tws93.

Example upgrade of an agent on IBM i systems

About this task

The following example shows the syntax used when using the **twsinst** script to upgrade an instance of the agent on IBM i system.

```
./twsinst -update
-uname TWS_user
-acceptlicense yes
-nobackup
-work_dir "/tmp/TWA/tws93"
```

Chapter 26. Uninstalling agents on IBM i systems

Learn how to uninstall agents on IBM i systems.

To uninstall IBM Workload Scheduler agents on an IBM i system using the **twinst** script, follow these steps:

1. Ensure that all IBM Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see *Administration Guide*.
2. Log on as QSECOFR and change your directory to */installation_dir/TWS*. For example: */home/user1/TWS* where *user1* is the name of IBM Workload Scheduler user.
3. From the Installation directory\TWS directory, run the **twinst** script as follows:

```
twinst -uninst -uname username [-wait minutes]  
[-lang lang_id] [-work_dir working_dir]
```

-uninst

Uninstalls IBM Workload Scheduler.

-uname username

The name of the user for which IBM Workload Scheduler is uninstalled. This user name is not the same as the user performing the installation logged on as QSECOFR.

-wait minutes

The number of minutes that the product waits for jobs that are running to complete before starting the uninstallation. If the jobs do not complete during this intervals the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-lang lang_id

The language in which the **twinst** messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

-work_dir working_dir

The temporary directory used for the IBM Workload Scheduler installation process files deployment. If you do not manually specify a path, the path is set to */tmp/TWA/tws93*.

The following example shows a **twinst** script that uninstalls the IBM Workload Scheduler agent, originally installed for **twsuser** user:

On IBM i systems:

```
./twinst -uninst -uname TWS_user
```

Chapter 27. The twsinst script log files on IBM i systems

About this task

The **twsinst** log file name is:

```
<TWS_INST_DIR>/twsinst_IBM_i_<TWS_user>^9.3.0.00.log
```

Where:

<TWS_INST_DIR>

The IBM Workload Scheduler installation directory. The default installation directory is /home/<TWS_user>.

<TWS_user>

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

Chapter 28. Analyzing return codes for agent installation, upgrade, restore, and uninstallation

Check how your operation completed by analyzing the return codes that are issued by twsinst.

Return codes that you can receive when you are installing, upgrading, restoring, or uninstalling agents. To analyze them and take corrective actions, run the following steps:

On Windows operating systems

1. Display the operation completion return code, by using the following command:
`echo %ERRORLEVEL%`
2. Analyze the following table to verify how the operation completed:

Table 53. Windows operating system agent return codes

Error Code	Description	User action
0	Success: The operation completed successfully without any warnings or errors.	None.
1	Generic failure	Check the messages that are displayed on the screen by the script. Correct the error and rerun the operation. If the error persists, search the IBM Support database for a solution at http://www.ibm.com/software/sysmgmt/products/support .
2	The installation cannot create the IBM Workload Scheduler user or assign the correct permission to it.	Verify the operating system policies and configuration. Verify the input values. If necessary, create the user manually before you run the installation.
3	The password is not correct or the installation cannot verify it.	Verify the operating system policies and configuration. Verify the input values.
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	Check the product system requirements at the following link: http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736 .
6	The IBM Workload Scheduler registry is corrupted.	Use the <code>recovInstReg</code> option to recover the registry. Then, rerun the operation.

Table 53. Windows operating system agent return codes (continued)

Error Code	Description	User action
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the localopts, the globalopts, the ita.ini, and the JobManager.ini files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

On UNIX and Linux operating systems:

1. Display the installation completion return code, by using the following command:
echo \$?
2. Analyze the following table to verify how the installation completed:

Table 54. UNIX or Linux operating system agent return codes

Error Code	Description	User action
0	Success: The installation completed successfully without any warnings or errors.	None.
1	Generic failure.	Check the messages that are displayed on the video by the script. Correct the error and rerun the operation. If the error persists, search the IBM Support database for a solution at http://www.ibm.com/software/sysmgmt/products/support .
2	The installation did not find the IBM Workload Scheduler user or its home directory. The IBM Workload Scheduler user that you specified either does not exist or does not have an associated home directory.	Verify the operating system definition of the IBM Workload Scheduler user.
3	Not applicable	
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	Check the product system requirements at the following link: http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736 .

Table 54. UNIX or Linux operating system agent return codes (continued)

Error Code	Description	User action
6	The IBM Workload Scheduler registry is corrupted.	Use the <code>recovInstReg</code> option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the <code>localopts</code> , the <code>globalopts</code> , the <code>ita.ini</code> , and the <code>JobManager.ini</code> files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

Part 7. Appendixes

Appendix A. Sample library (SEQQSAMP)

The SEQQSAMP library contains samples to help you install, migrate, and customize IBM Workload Scheduler for z/OS.

In most cases, you need only add installation-specific JCL to adapt a member in SEQQSAMP to your requirements. Appendix A, “Sample library (SEQQSAMP)” lists all members in the SEQQSAMP library and provides a brief description of each member. The following pages describe the samples relating to installing IBM Workload Scheduler for z/OS in more detail. Descriptions of other sample-library members are included in the book that describes the function demonstrated by the sample. For example, program-interface samples are described in *Tivoli Workload Automation: Developer’s Guide: Driving IBM Workload Scheduler for z/OS*.

Some of the samples provided address a specific function and you might be able to use the sample unchanged in your environment. If you need to change a sample member, it is advisable to copy the source to a separate library. The original sample member is then available for reference. It is also recommended that you create an SMP/E *usermod* for each sample member you run in the production environment. Changes to the sample source code will then be flagged for your attention, and subsequent updates can be reflected in the production code as soon as possible.

Table 55. SEQQSAMP library members

Member	Brief description
EQQ9RF01	Sample RACF router table entry to enable security environment.
EQQ9RFDE	Sample RACF class descriptor entry to enable security environment.
EQQ9SM01	JCL to install RACF router table update.
EQQ9SMDE	JCL to install RACF class descriptor update.
EQQACPT x	Sample SMP/E ACCEPT JCL for the controller software, where the value of x depends on the language.
EQQACTR1	Sample SMF exit IEFACTRT, written in assembler, to enable job-tracking.
EQQAIXST	Parameters used by the EQQX9AIX and EQQAIXTR samples.
EQQAIXTR	Sample tracker running on AIX®, used with EQQX9AIX.
EQQALLOC	JCL to allocate the IBM Workload Scheduler for z/OS distribution and target libraries.
EQQALSMP	Sample JCL to allocate and initialize the SMP/E environment needed to install IBM Workload Scheduler for z/OS
EQQAPISM	ASCII file containing a sample API application.
EQQAPPL x	Sample SMP/E APPLY JCL for the controller software, where the value of x depends on the language.
EQQAUDIB	Sample to invoke EQQAUDIT in batch mode outside of the dialog, processing EQQTROUT or EQQDROUT data set. Note: EQQAUDIB can be used successfully only if the EQQTROUT dsname and the EQQAUDIT output dsn fields in the EQQJOBSA panel are typed out.
EQQBENCO	Sample JCL that encrypts the password defined in the OSLCOPTS initialization statement used to configure IBM Workload Scheduler for z/OS to integrate with OSLC.
EQQBENCR	Sample EQQE2EPW JCL to run the utility that encrypts the Windows passwords set in the USRPSW parameter of the USRREC statements.

Table 55. SEQQSAMP library members (continued)

Member	Brief description
EQQBSCAN	Batch loader sample to validate an application description.
EQQBSubS	Batch loader sample to create four application descriptions and two operator instructions. Output is directed to a subsystem.
EQQBVSAM	Batch loader sample to create an application description and two operator instructions. Output is directed to a VSAM data set that is allocated by the sample.
EQQCHKEV	A sample JCL to display EQQTWSIN and EQQTWSOU event data set content information.
EQQCLEAN	Sample procedure invoking EQQCLEAN program.
EQQCONOP	Sample parameters used by EQQCONO.
EQQCONO	Sample started task procedure for controller only.
EQQCONP	Sample initial parameters for a controller and tracker in the same address space.
EQQCON	Sample started task procedure for a controller and tracker in the same address space.
EQQCVM2	Sample to enable submission and tracking on VM systems using EQQUX009.
EQQCVM	Sample to enable job-tracking facilities on VM systems.
EQQDBENC	Contains the JCL to encrypt the password in the DBOPT statement.
EQQDBOPT	Sample DBOPT statement.
EQQDDDEF	Sample job to allocate DDDEFs in SMP/E.
EQQDELDI	JCL and usage notes for the data set deletion function.
EQQDLFX	Assembler installation sample of DLF connect/disconnect exit.
EQQDPCOP	JCL and usage notes for copy VSAM function.
EQQDPX01	DP batch sample user exit to update the scheduling environment.
EQQDSCL	Batch Clean Up sample.
EQQDSCLP	Batch Clean up sample parameters.
EQQDSECT	Assembler version of PIF data areas.
EQQDSEX	Batch Export sample.
EQQDSEXP	Batch Export sample parameters.
EQQDSIM	Batch Import sample.
EQQDSIMP	Batch Import sample parameters.
EQQDSRG	Batch sample reorg.
EQQDSRI	Batch Recovery index.
EQQDSRIP	Batch Recovery index parameters.
EQQDST	Sample procedure to start Data Store.
EQQDSTP	Parameters for sample procedure to start Data Store.
EQQE2EP	Sample initial parameters for server and batch to define if the end-to-end scheduling with fault tolerance capabilities is active.
EQQICNVH	Sample job to migrate history DB2 tables.
EQQICNVS	Sample job to migrate VSAM files.
EQQINIDB	Sample to create the history database.
EQQISMKD	Sample job to run EQQMKDIR exec for directories.
EQQJCCTB	JCL to assemble a JCC message table macro definition.
EQQJCLIN	Sample JCL to start program EQQPDFL.
EQQJER2U	Sample to restore the EXIT7 as a JES2 usermod.

Table 55. SEQQSAMP library members (continued)

Member	Brief description
EQQJER2V	Sample to restore the EXIT5 as a JES2 usermod.
EQQJER3U	Sample to restore the EQQUX191 and EQQUX291 as JES3 usermods.
EQQJES21	JCL to assemble and link-edit the JES2 EXIT51.
EQQJES2	JCL to assemble and link-edit a JES2 exit.
EQQJES2U	JCL to install the JES2 EXIT7 as an SMP/E usermod.
EQQJES2V	JCL to install the JES2 EXIT51 as an SMP/E usermod.
EQQJES3	JCL to assemble and link-edit a JES3 exit.
EQQJES3U	JCL to install a JES3 exit as an SMP/E usermod.
EQQJVXIT	Sample assembler JCL-variable-substitution exit. Also used for variable substitution in System Automation commands.
EQQLSJCL	Sample JCL to invoke the EQQLSENT macro.
EQQMKDIR	Sample exec to create directories.
EQQNCFACT	Sample parameters for an SNA connection between controller and tracker.
EQQNETW1	REXX EXEC that receives IBM Workload Scheduler for z/OS WTO messages and issues MVS commands.
EQQNETW2	PL/I NetView command processor that uses EQQUSINT to change the status of operations.
EQQNETW3	REXX EXEC that uses EQQEVPGM to change the status of operations.
EQQOCWTO	Sample job to assemble and linkedit the IPOWTO routine used by the PIF REX sample.
EQQORST	Resets the USS environment for the end-to-end scheduling with fault tolerance capabilities.
EQQOS2ST	Parameters used by the EQQX9OS2 and EQQOS2TR samples.
EQQOS2TR	Sample tracker running on OS/2, used with EQQX9OS2.
EQQPCS01	Allocates data sets that need to be unique within the SYSPLEX.
EQQPCS02	Allocates data sets that need to be unique to each MVS image in the SYSPLEX.
EQQPCS03	Generates a job that allocates VSAM copy data sets.
EQQPCS04	Defines Data Store VSAM files and initializes them.
EQQPCS05	Allocates files used by a controller to enable fault-tolerant workstations.
EQQPCS06	Allocates VSAM data sets for integration with the end-to-end scheduling with fault tolerance capabilities.
EQQPCS07	Allocates VSAM data sets for Restart and Cleanup.
EQQPCS08	Allocates USS files for Java utilities enablement.
EQQPCS09	Allocates the GDG root and VSAM data set used as input by the archiving process supporting the Dynamic Workload Console reporting feature.
EQQPCS10	Creates the SSL work directory used for TCP/IP communication with the controller.
EQQPCS11	Allocates data sets (EQQOUCFV and EQQOUCFKP) used for the retrieval of job logs in the z-centric environment with the Output collector.
EQQPCS12	Allocates the GDG root to archive the MLOG files.
EQQPIFAD	Program-interface PL/I sample that creates a two-operation application in the AD database.
EQQPIFAP	Program-interface PL/I sample that resolves JCL variables.
EQQPIFCB	Program-interface assembler samples for various current plan or long-term plan actions.
EQQPIFCL	Program-interface assembler sample that uses the DAYSTAT command to return work or free status for a particular date.
EQQPIFDJ	Program-interface assembler sample, deletes JCL for completed occurrences from JS data set.

Table 55. SEQQSAMP library members (continued)

Member	Brief description
EQQPIFJC	Program-interface COBOL sample to manipulate JCL variable tables.
EQQPIFJD	Program-interface PL/I sample that can either list or delete records in the JCL repository data set (JS).
EQQPIFJV	Program-interface PL/I sample to manipulate JCL variable tables.
EQQPIFJX	Sample to maintain the JCL repository.
EQQPIFOP	Program-interface REXX sample to modify an operation in the current plan.
EQQPIFPR	Program-interface REXX sample to list all cyclic periods.
EQQPIFWI	Program-interface PL/I sample to modify capacity values in an open interval of a current plan workstation.
EQQPMCKP	Merges the checkpoint data sets of the old and new systems in the production system migration process. See "Produce a checkpoint data set containing data from the old production system" on page 202.
EQQPROC	Sample procedure, started by IBM Workload Scheduler for z/OS, to initiate purge of DLF objects.
EQQRECV x	Sample SMP/E RECEIVE JCL for the controller software, where the value of x depends on the language.
EQQREPRO	Is invoked by EQQSMLOG to copy the contents of the outgoing MLOG file onto the GDG data set. You must copy this sample to the PARMLIB of the controller.
EQQRETWT	Sample program to simulate abends, return codes and waits.
EQQRMDS	Usage notes for the job-log-retrieval exit object code to interface to RMDS.
EQQRXSTG	An assembler routine to get and free storage for the REXX program-interface samples.
EQQSAMPI	JCL to load sample data for application descriptions, operator instructions, and workstation descriptions to the databases.
EQQSERP	Sample initial parameters for a Server.
EQQSER	Sample started task procedure for a Server.
EQQSLCHK	JCL to perform a syntactic check on SCRIPT library members. Note: EQQSLCHK sample JCL is generated with the following DD card: EQQMLOG DD SYSOUT=*" <p>If the EQQMLOG ddname is associated with a physical data set that has not sufficient size , a D37 abend followed by a user abend U4036 might occur. In this case, you must reallocate the EQQMLOG data set with more space.</p> <p>To re-create a new EQQSLCHK sample JCL, run again option 1 (Create sample job JCL) of EQQJOBS.</p>
EQQSMF	JCL to assemble and install the SMF exits.
EQQSMLOG	Sample procedure that creates the GDG data set where the outgoing MLOG file is archived when the MLOG switching function takes effect. Uses the EQQREPRO input parameter.
EQQTCPCT	Sample definitions for TCP/IP communication between tracker and controller.
EQQTRAP	Sample initial parameters for a Tracker.
EQQTRA	Sample started task procedure for a Tracker.
EQQTROPT	Sample TRGOPT statement.
EQQU831	Sample SMF exit IEFU83 to enable job tracking and optionally include data set triggering support.
EQQUJI1	Sample SMF exit IEFUJI to enable job-tracking.

Table 55. SEQQSAMP library members (continued)

Member	Brief description
EQQUSIN1	EQQUSIN subroutine sample to change the status of an operation.
EQQUSIN2	EQQUSIN subroutine sample to change the availability of a special resource.
EQQUSIN3	EQQUSIN subroutine sample to change the status of a workstation.
EQQUSIN4	EQQUSIN subroutine sample to backup an IBM Workload Scheduler for z/OS resource data set.
EQQUSIN5	EQQUSIN subroutine sample to update the USERDATA field of an operation.
EQQUX001	Sample job-submit exit.
EQQUX002	Sample job-library-read exit.
EQQUX004	Sample event-filtering exit.
EQQUX011	Sample job-tracking log write exit.
EQQUX013	Sample job-tailoring prevention exit.
EQQUX0N	Sample PL/I start/stop exit, EQQUX000.
EQQUX191	Sample JES3 exit IATUX19 to enable job tracking.
EQQUX291	Sample JES3 exit IATUX29 to enable job tracking.
EQQUX9N	Sample PL/I operation-initiation exit, communicating with VM (EQQUX009).
EQQUXCAT	Sample restart and clean up exit for the EQQCLEAN program.
EQQUXPIF	Sample user exit to validate application descriptions.
EQQXSAZ	Sample assembler system command exit, communicating with System Automation invoked in place of EQQUX007 for automation workstations.
EQQVTAMN	Sample VTAM definition for SNA connection between tracker and controller.
EQQVTAMS	Sample VTAM definition for server SNA connection.
EQQX5ASM	Sample SYSOUT archiving exit.
EQQX6ASM	Sample incident-record-create exit.
EQQX6JOB	Sample batch-job skeleton JCL used by EQQX6ASM.
EQQX7ASM	Sample change-of-status exit.
EQQX7JOB	Sample batch-job skeleton JCL used by EQQX7ASM.
EQQX9AIX	Sample assembler operation-initiation exit, communicating with AIX.
EQQX9OS2	Sample assembler operation-initiation exit, communicating with OS/2.
EQQXCFC	Sample definitions for XCF connection between tracker and controller.
EQQXIT51	Sample JES2 EXIT51 to enable job tracking for JES2 with z/OS version 1 release 7, and later.
EQQXIT74	Sample JES2 EXIT7 to enable job tracking for JES2 level version 4 release 1 and later.
EQQXML01	Sample XML file for data set triggering event rule definitions.
EQQYCBAG	Sample to unload a group application (and all the applications belonging to it) into a sequential file in Batch Loader Control statement format.
EQQYCBAT	Run the Batch Command Interface tool.
EQQYRJCL	Sample JCL to run the Control Language tool.
EQQYRMSG	Messages used by the Control Language tool.
EQQYRPRC	Sample procedure to run the Control Language tool.
EQQYRPRM	Sample initialization parameter file for the Control Language tool.

Using the Visual Age compiler

With the z/OS operating system, the Visual Age PL/I compiler replaces all the previous PL/I compilers. Therefore, if you use this compiler, you need to customize the samples in PL/I as follows:

1. Replace the PL/I compiler invocation statement:

```
EXEC PGM=IEL0AA
```

with:

```
EXEC PGM=IBMZPLI
```

2. Link into a PDS/E data set for SYSLMOD or include a pre-link edit step in the JCL.

As an example, here is the JCL for the EQQPIFJV sample using the Visual Age PL/I compiler:

```
//EQQPIFJV JOB MSGCLASS=N, .....
//PLI1 EXEC PGM=IBMZPLI,REGION=1024K,
// PARM='OBJECT,OPTIONS'
//STEPLIB DD DSN=IBMZ.V2R2M1.SIBMZCMP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(,PASS),
// DSN=&&OBJ1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SYSIN DD *
/*
/*
//PLI2 EXEC PGM=IBMZPLI,REGION=1024K,
// COND=(4,LT,PLI1),PARM='OBJECT,OPTIONS'

//SYSPRINT DD SYSOUT=*
//STEPLIB DD DSN=IBMZ.V2R2M1.SIBMZCMP,DISP=SHR
//SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(2,1)),
// DISP=(,PASS),DSN=&&OBJ2
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SYSIN DD *
.....
/*
/*
//*****
//* PRE-LINK-EDIT STEP *
//*****
//PLKED EXEC PGM=EDCPRLK,COND=(8,LT,PLI1),
// REGION=2048K
//SYSDEFSD DD DSN=&&DEF1,LRECL=80,BLKSIZE=3200,
// DISP=(,PASS)
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR
//SYMSGS DD DSN=CEE.SCEEMSGP(EDCPMSGE),DISP=SHR
//SYSLIB DD DUMMY
//SYSMOD DD DSN=&&PLNK,DISP=(,PASS),
// UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN DD DSN=&&OBJ1,DISP=(OLD,DELETE)
// DD DSN=&&OBJ2,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//*****
//* SCEELKED ADDED TO SYSLIB ON LINK STEP *
//*****
//LKED EXEC PGM=IEWL,PARM='XREF',
// COND=(4,LT,PLI2),REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DISP=SHR,DSN=CEE.SCEELKED
// DD DISP=SHR,DSN=USER.OPC23.LINKLI
```

```

//SYSLMOD DD DISP=SHR,DSN=SVIOLA.SEQQLMD0
//OPCLIB DD DISP=SHR,DSN=USER.OPC23.LINKLI
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SEQOBJ1 DD DISP=(OLD,DELETE),DSN=&&PLNK
//SYSLIN DD *
INCLUDE SEQOBJ1
INCLUDE OPCLIB(EQQYCOM)
SETCODE AC(1)
ENTRY CEESTART
NAME EQPIFT(R)
/*
/**
//EQPIFT EXEC PGM=EQPIFT,PARM='NOSTAE,NOSPIE',
// COND=(4,LT,LKED), REGION=4096K
//STEPLIB DD DISP=SHR,DSN=SVIOLA.SEQQLMD0
// DD DISP=SHR,DSN=USER.OPC23.LINKLIB
//EQQLIB DD DSN=EQQ.V2R3M0.SEQQMSG0,DISP=SHR
//EQQYPARM DD DISP=SHR,DSN=XXXX.YYYY.ZZZZ(YPARM)
//EQQMLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//EQQDUMP DD SYSOUT=*
//EQQMSG DD SYSOUT=*
//CARDIN DD *
.....
/*
/**

```

SMP/E samples

The following SEQQSAMP members relate to SMP/E processes.

Environment setup

You can use the sample library members EQQALSMP, EQQDDDEF, and EQQALLOC to create and initialize the SMP/E environment and the IBM Workload Scheduler for z/OS product libraries that are needed to support the installation and continuing maintenance of IBM Workload Scheduler for z/OS.

The EQQALSMP job performs the following functions:

- Initializes an SMP/E CSI, adding a global zone, and the IBM Workload Scheduler for z/OS FMID.

The EQQDDDEF job sets up DDDEFs for all IBM Workload Scheduler for z/OS data sets to provide for basic JCL requirements for RECEIVE, APPLY, and ACCEPT processing.

The EQQALLOC job allocates all Tivoli target and distribution libraries. The JCL also contains a number of steps, which are currently commented out. You can use those steps to delete the IBM Workload Scheduler for z/OS libraries if you need to reinstall the product.

RECEIVE processing

The sample library members EQQRECVE, EQQRECVS, EQQRECJV, EQQRECVD, and EQQRECVK contain JCL that you can use to run SMP/E RECEIVE processing for IBM Workload Scheduler for z/OS data sets. These library members enable you to receive the following IBM Workload Scheduler for z/OS features:

- Tracker
- Controller
- z/OS connector enable
- End-to-end and Java enabler

Each of these jobs performs RECEIVE processing for a particular NLS feature:

EQQRECVE

Receives all the scheduler base and tracker components plus the English feature for the controller.

EQQRECVS

Receives all the scheduler base and tracker components plus the Spanish feature for the controller.

EQQRECJV

Receives all the scheduler base and tracker components plus the Japanese feature for the controller.

EQQRECVD

Receives all the scheduler base and tracker components plus the German feature for the controller.

EQQRECVK

Receives all the scheduler base and tracker components plus the Korean feature for the controller.

You might need to change the distribution library and zone name to reflect those defined in the IBM Workload Scheduler for z/OS CSI.

For further details, see the IBM Workload Scheduler for z/OS program directory.

Note: The program directory refers to trackers as *agents*, and to the controller as the *engine*.

APPLY processing

The sample library members EQQAPPLE, EQQAPPLS, EQQAPPLJ, EQQAPPLD, and EQQAPPLK contain JCL that you can use to run SMP/E APPLY processing for IBM Workload Scheduler for z/OS. These members enable you to apply the following features:

- Tracker
- Controller
- z/OS connector enable
- End-to-end and Java enabler

Each of these jobs performs APPLY processing for a particular NLS feature:

EQQAPPLE

Applies all the scheduler base and tracker components plus the English feature for the controller.

EQQAPPLS

Applies all the scheduler base and tracker components plus the Spanish feature for the controller.

EQQAPPLJ

Applies all the scheduler base and tracker components plus the Japanese feature for the controller.

EQQAPPLD

Applies all the scheduler base and tracker components plus the German feature for the controller.

EQQAPPLK

Applies all the scheduler base and tracker components plus the Korean feature for the controller.

You might need to change the distribution library and zone name to reflect those defined in the IBM Workload Scheduler for z/OS CSI.

For further details, see the IBM Workload Scheduler for z/OS program directory.

Note: The program directory refers to trackers as *agents*, and to the controller as the *engine*.

ACCEPT processing

The sample library members EQQACPTE, EQQACPTS, EQQACPTJ, EQQACPTD, and EQQACPTK contain JCL that you can use to run SMP/E ACCEPT processing for IBM Workload Scheduler for z/OS. These members enable you to accept the following features:

- Tracker
- Controller
- z/OS connector enable
- End-to-end and Java enabler

Each of these jobs performs ACCEPT processing for a particular NLS feature:

EQQACPTE

Accepts all the scheduler base and tracker components plus the English feature for the controller.

EQQACPTS

Accepts all the scheduler base and tracker components plus the Spanish feature for the controller.

EQQACPTJ

Accepts all the scheduler base and tracker components plus the Japanese feature for the controller.

EQQACPTD

Accepts all the scheduler base and tracker components plus the German feature for the controller.

EQQACPTK

Accepts all the scheduler base and tracker components plus the Korean feature for the controller.

You might need to change the distribution library and zone name to reflect those defined in the IBM Workload Scheduler for z/OS CSI.

For further details, see the IBM Workload Scheduler for z/OS program directory.

Note: The program directory refers to trackers as *agents*, and to the controller as the *engine*.

SMF exits

The following text provides details of the SEQQSAMP members relating to SMF exits.

Note: If version ASMA90 of the compiler reports errors, and the RMODE=ANY statement is defined, remove the RMODE=ANY statement from the sample exit.

Exit installation

The sample library member EQQSMF contains the JCL needed to assemble the SMF exits required for IBM Workload Scheduler for z/OS. The job also defines an SMP/E usermod to connect the SMF exits to your target zone.

A single usermod is used to define the three SMF exits. You can, if you prefer, define usermods for each exit.

To restore the JES exits as SMP/E usermods, use the samples EQQJER2U, EQQJER2V, and EQQJER3U.

Job step termination exit

The sample library member EQQACTR1 contains the assembler source code of an SMF job/step termination exit, IEFACTRT. The sample contains two subroutines:

- OPCASUB provides the necessary IBM Workload Scheduler for z/OS code to track job- and step-end events.
- LOCALSUB generates WTO messages for step- and job-end.

If you use the IBM Workload Scheduler for z/OS Restart and Cleanup functionality or other functions, such as the one related to the NOERROR table (described in the *Customization and Tuning* manual), you are required to install this exit. Use the sample provided with the product to install this exit.

From the introduction of the usability enhancement on, the IEFACTRT exit creates two different tables in the joblog, the Steptable and the Not_Executed_Step_Table.

If you use the IBM Workload Scheduler for z/OS Restart and Cleanup functionality or other functions, such as the one related to the NOERROR table, and you want to replace this subroutine with your own, you need to comply with the following restrictions:

- The fields JOBNAME, STEPNAME, PROCSTEP and STEPNO must continue to be filled on the basis of the following logic:
 - JOBNAME must contain the name of the job
 - STEPNAME is the label of the EXEC PROC=... Card and must be filled only if a PROC is used
 - PROCSTEP is the label of the EXEC PGM=... Card and must be filled also if a PROC is not used
 - STEPNO must contain the sequence number of the steps inside the job
- The JOBNAME, STEPNAME, PROCSTEP identifiers in the tables header must match the values specified in the *HDRJOBNAME*, *HDRSTEPNAME*, and *HDRPROCNAME* parameters of the DSTOPTS DATASTORE statement.
- The layout of STEPTABLE and NOT_EXECUTED_STEP_TABLE must be in compliance with the following rules:
 - JOBNAME must be preceded by a hyphen sign (-), some characters can be inserted between a hyphen sign and the JOBNAME
 - JOBNAME must be followed by a blank
 - STEPNAME must be preceded and followed by a blank
 - PROCSTEP must be preceded and followed by a blank

- STEPNO must be preceded by a blank
- STEPNO must follow the PROCSTEP in the NOT_EXECUTED_STEP_TABLE
- NOT_EXECUTED_STEP_TABLE must be aligned to STEPTABLE as far as it concerns JOBNAME, STEPNAME and PROCSTEP information.
- The string "JOBXXXX ENDED. NAME-" must be aligned so that the JOBNAME JOBXXXX is under the JOBNAME header.
- JOBNAME, STEPNAME, PROCSTEP position in the STEPTABLE and in the NOT_EXECUTED_STEP_TABLE must match the values specified in the HDRJOBLENGTH, HDRSTEPLength, HDRPROCLength parameters of the DSTOPTS DATASTORE statement. STEPNO position in the STEPTABLE must match the value specified in HDRSTEPNOLENGTH.
- User-customized records issued in the STEPTABLE and in the NOT_EXECUTED_STEP_TABLE must be avoided.

Initialization exit

The sample library member EQQUJ11 contains the assembler source code of an SMF initialization exit, IEFUJ1. IBM Workload Scheduler for z/OS uses events generated from the exit to track job start information.

If your installation is already using an IEFUJ1, incorporate the code into your existing exit and reassemble.

Record write exits

The sample library member EQQU831 contains the assembler source code of a record write exit, IEFU83. IBM Workload Scheduler for z/OS uses events generated from the exit to track print group and purge information.

If your installation is already using an IEFU83, incorporate the code into your existing exit and reassemble.

You can optionally include support for both the data set triggering and job-tracking functions using the EQQU831 sample. This provides you with a method to automatically generate a special resource availability depending on specific actions affecting data sets. The event can be used by IBM Workload Scheduler for z/OS to change the status of a special resource to make it available for operations and/or to trigger an application to be added to the current plan. You specify the data sets you want special resource availability events for using a specific macro, as described in Appendix D, "Invoking the EQQLSENT macro," on page 391. See "Implementing support for data set triggering" on page 92 for more information about data set triggering. Use the EQQSMF sample to install EQQU831.

If you do not track print operations through IBM Workload Scheduler for z/OS, and you do not want to include data set triggering support, you need not change IEFU83.

JES exits

The following text provides details of the SEQQSAMP members relating to JES exits.

Note: If version ASMA90 of the compiler reports errors, and the RMODE=ANY statement is defined, remove the RMODE=ANY statement from the sample exit.

Exit installation

The sample library contains a number of members to assemble and link-edit JES exits. EQQJES2, EQQJES21, and EQQJES3 provide sample JCL to assemble and link-edit of JES2 and JES3 exits respectively. However, it is recommended that you use members EQQJES2U, EQQJES2V, and EQQJES3U. These samples provide the JCL to install the JES exits as SMP/E usermods. The usermods are defined so that both the JES and IBM Workload Scheduler for z/OS target zones are informed of the dependencies. This ensures that future maintenance, to either the JES component or the IBM Workload Scheduler for z/OS component, will be handled correctly.

JES2 QMOD phase change exit

The sample library member EQQXIT51 contains the assembler source code of the JES2 QMOD Phase Change exit, JES EXIT51. This sample must be used with z/OS 1.7 or later. IBM Workload Scheduler for z/OS uses JES2 EXIT51 to detect job errors occurring during the JES2 input phase, and to trigger the creation of IJ2 events for started task.

JES2 JCT I/O exit

The sample library member EQQXIT74 contains the assembler source code of a JES2 JCT I/O exit, JESEXIT7. EQQXIT74 is used for JES2. IBM Workload Scheduler for z/OS uses JESEXIT7 to detect new jobs on the internal reader and also to detect output group purge.

If you are already using a JESEXIT7, and want to keep the IBM Workload Scheduler for z/OS job-tracking support in a separate load module, you can specify that JES use multiple EXIT7 modules in your JES2 parameters.

JES3 OSE modification exit

The sample library member EQQUX191 contains the assembler source code of a JES3 OSE modification exit, IATUX19. IBM Workload Scheduler for z/OS uses events generated from the exit to detect output group purge.

If you are already using an IATUX19, you should include the code in your existing exit and reassemble.

Note: If you are using JES3 Exit IATUX72 then this exit must return with R15 = 8 to call IATUX19.

JES3 input service final-user exit

The sample library member EQQUX291 contains the assembler source code of a JES3 input service final-user exit, IATUX29. IBM Workload Scheduler for z/OS uses events generated from the exit to detect new jobs on the internal reader.

If you are already using an IATUX29, then you should incorporate the code into your existing exit and reassemble.

RACF samples

The following text provides details of the SEQQSAMP members relating to RACF changes, which are required for IBM Workload Scheduler for z/OS security.

Class descriptor table

The sample library member EQQ9RFDE provides the class descriptor entry required to define the IBM Workload Scheduler for z/OS security environment to RACF, or a functionally equivalent product.

Each class descriptor contains control information needed by RACF to validate class names and is a CSECT in the load module ICHRRRCDE.

You can use member EQQ9SMDE to install ICHRRRCDE as an SMP/E usermod on the RACF target zone.

Router table

The sample library member EQQ9RF01 provides the router table entry required to define the IBM Workload Scheduler for z/OS security environment to RACF, or a functionally equivalent product.

This is a sample RACF router table that provides action codes to determine if RACF is invoked on behalf of the RACROUTE macro.

You can use member EQQ9SM01 to install ICHRRFR01 as an SMP/E usermod on the RACF target zone.

Sample library (SEQQSAMP)

The EQQYCBAG member of the EQQSAMP library provides a sample in which the Batch Command Interface Tool (BCIT) is used to unload a group application, and all applications belonging to it, into a sequential file in batchloader control statement format.

The group applications, as well as other applications, can be modified via the batchloader control statements. From then on, you can use the sequential file as input for the batchloader run.

This sample consists of two jobs:

1. The unload job, that uses the batch command interface tool
2. The load job, that uses the batchloader.

Before running the job, you need to customize it with correct values for the job card name, data set names, subsystem name, and so on.

The EQQBENCR member of the EQQSAMP library provides a sample of the EQQE2EPW JCL that you can use to encrypt the passwords written in plaintext in the USRREC statement of the USRINFO configuration member, or to insert additional USRREC statements through the SYSIN data.

Following is an example of the sample EQQE2EPW JCL.

```
//EQQE2EPW EXEC PGM=EQQUPTOP,REGION=64M,TIME=1440
//*****
//* THIS IS A SAMPLE JCL TO ENCRYPT THE PASSWORDS IN THE USRREC      *
//* STATEMENT CONTAINED IN THE EQQPARM LIBRARY MEMBER AS SPECIFIED *
//* BY THE USRMEM KEYWORD IN THE TOPOLOGY STATEMENT, FOR EXAMPLE  *
//* USRMEM(USRINFO).                                               *
//* THE TWS FOR ZOS DEFAULT FOR THIS MEMBER NAME IS USRINFO, AS   *
//* DEFINED IN THE EQQE2EP INSTALLATION SAMPLE.                   *
//* SPECIFY THE LIBRARY THAT CONTAINS THE USRINFO MEMBER,        *
//* INCLUDING THE MEMBER NAME, IN THE EQQUSRIN DD OF THIS JCL.     *
```

```

/** SPECIFY IN THE SYSIN DD EITHER THE NAME OF A DATA SET (INCLUDING *
/** THE MEMBER NAME, IF PDS) CONTAINING THE USRREC STATEMENTS OR THE *
/** USRREC STATEMENTS DIRECTLY AS INLINE PARAMETERS. THESE ARE THE *
/** USRREC STATEMENTS THAT YOU WOULD LIKE TO ADD TO THE USRINFO *
/** DATASET MEMBER. *
/** NOTICE THAT ALL THE THREE KEYWORDS OF THE USRREC STATEMENT *
/** (USRCPU, USRNAM, USRPSW) ARE REQUIRED IN THE SYSIN. INSERT ONE *
/** USRREC STATEMENT KEYWORD PER ROW. *
/** FOR EXAMPLE: *
/**     SYSIN DD * *
/**         USRCPU(WS01) *
/**         USRNAM('TEST1') *
/**         USRPSW('ABC123') *
/**         USRCPU(WS02) *
/**         USRNAM('TEST2') *
/**         USRPSW('EFG567') *
/** AS RESULT THE PASSWORDS SPECIFIED IN THE USRPSW KEYWORDS WILL BE *
/** ENCRYPTED (EITHER IF IN THE SYSIN OR IN THE USRINFO MEMBER) AND *
/** THESE USRREC STATEMENTS ARE STORED IN THE USRINFO DATA SET AS *
/** SPECIFIED IN THE EQQUSRIN DD CARD. *
/** IF THE SYSIN IS NOT SPECIFIED, OR SPECIFIES DUMMY, ONLY THE *
/** PASSWORDS PRESENT IN THE USRINFO DATA SET WILL BE ENCRYPTED. *
/** NOTE: *
/** REVIEW ALL THE JCL CONTENT AND SETTINGS *
/** BEFORE SUBMITTING THIS JCL DOUBLE CHECK THAT THE DD DEFINITIONS *
/** SUIT YOUR INSTALLATION. MAINLY DOUBLE CHECK THAT EQQUSRIN AND *
/** SYSIN DD ARE PROPERLY DEFINED. CHECK THE JOB REGION SIZE. *
/** ADD YOUR JOB CARD. CONSIDER BACKING UP YOUR PARMLIB DATASET. *
/*******
//EQQE2EPW EXEC PGM=EQQUPTOP,REGION=64M,TIME=1440
//STEPLIB DD DISP=SHR,DSN=&STEPDSN
//EQQMLOG DD SYSOUT=&FPCLA
//EQQMLIB DD DISP=SHR,DSN=&MSGLIB
//EQQPARM DD DISP=SHR,DSN=&PARMDSN
//EQQDUMP DD SYSOUT=&FPCLA
//SYSUDUMP DD SYSOUT=&FPCLA
//EQQUSRIN DD DISP=SHR,DSN=&PARMDSN(USRINFO)
//SYSIN DD DISP=SHR,DSN=dsname(member_name)
/**SYSIN DD *
/**USRCPU(TEST)
/**USRNAM('DUMMY')
/**USRPSW('ABC123')

```

Note:

1. Insert the keywords contained in SYSIN, either inline or in a data set, one per row.
2. The keywords are the same as the ones used for the USRREC statement: USRCPU, USRNAM, and USRPSW. These three keywords are all required in the SYSIN.
3. Rows containing only comments are inserted into the USRINFO data set member (pointed by the EQQUSRIN DD card) as they are, starting at column 13.
4. You can write comments on every row, but, depending on the statement length, they can result truncated or be overwritten by the row content. The suggested range is from column 50 to column 60. Rows containing only comments are allowed.
5. The password length after the encryption is always 31 bytes and the statements start at column 13, therefore you can use maximum 60 characters per row.
6. During the data set scanning process, if duplicated USRREC statements (same values for USRCPU and USRNAM) are found, the last USRREC found is inserted and the first USRREC is removed. The scan is performed from the top

of the data set. The statements contained in SYSIN are considered more recent compared with the statements in USRINFO.

7. A light syntax checking is performed on the SYSIN data set content. Only few checks on the USRINFO data set. A complete syntax checking is performed on the content of the final USRINFO data set, when a DP batch or Symphony renew is performed, as usual.
8. New rows added to the USRINFO member are flagged with the /*JADD*/ comment starting at column 73. This will help to locate the modified lines. Following is an example of how the inserted rows look like:

```
USRREC USRCPU(WS01)                               /*JADD*/
        USRNAM('TEST01')                          /*JADD*/
        USRPSW('M($H7ggTDê;Dê ā ä7LN};ôā°Nä)}| î') /*JADD*/
```

Remove the /*JADD*/ flag manually before the next run of job EQQE2EPW, to distinguish the new lines that will be added. To do this, edit the USRINFO member and do one of the following actions:

- Remove manually the comments /*JADD*/ using the edit command CUT of the host emulator.
- Use the TSO edit command CHANGE ALL:
CHANGE '/*JADD*/' ' ' ALL
- Use the TSO edit commands RENUM and UNNUM.

Appendix B. Configuration examples

This appendix gives examples of IBM Workload Scheduler for z/OS configuration. The examples are based on z/OS JES2, but they are also valid for z/OS JES3 systems, or a combination of JES2 and JES3 systems. Each example shows:

- The controlling system, with the controller and the tracker started in separate address spaces
- All IBM Workload Scheduler for z/OS address spaces as IBM Workload Scheduler for z/OS systems
- A summary of actions that the workload restart function could take automatically
- Sample initialization statements that you can use to create the configuration
- The IBM Workload Scheduler for z/OS components that are required, the flow of automatic work submission, and event collection in various system combinations.

The controlling system

The controlling system is shown in the examples only with the controller and the tracker connected via either shared DASD or XCF. But you can connect them via NCF or TCP/IP, if you prefer this method.

IBM Workload Scheduler for z/OS can support remote systems that are in different time zones from the controlling system. For more information on time zone support and daylight saving time changes, see *IBM Workload Scheduler for z/OS: Managing the Workload*.

Automatic restart actions

The possible actions vary according to the type of connection between the controller and the tracker.

Initialization statements

Default values are used for statements that do not specifically relate to the configuration. The statements are specified in one or more parameter library members.

Multi-access spool systems connected through shared DASD

Figure 28 on page 372 shows two z/OS JES2 multi-access spool (MAS) complexes that are connected through shared DASD.

Systems A and B form a MAS complex. System A is the IBM Workload Scheduler for z/OS controlling system. It shares spool with System B, which is a controlled IBM Workload Scheduler for z/OS system. Work is sent directly to this complex by the controller on System A. The work is processed on one of these two systems, depending on installation parameters. You represent this complex to IBM Workload Scheduler for z/OS by defining a computer workstation with a blank destination field. That is, all work for this workstation is submitted to the system that the controller is started on.

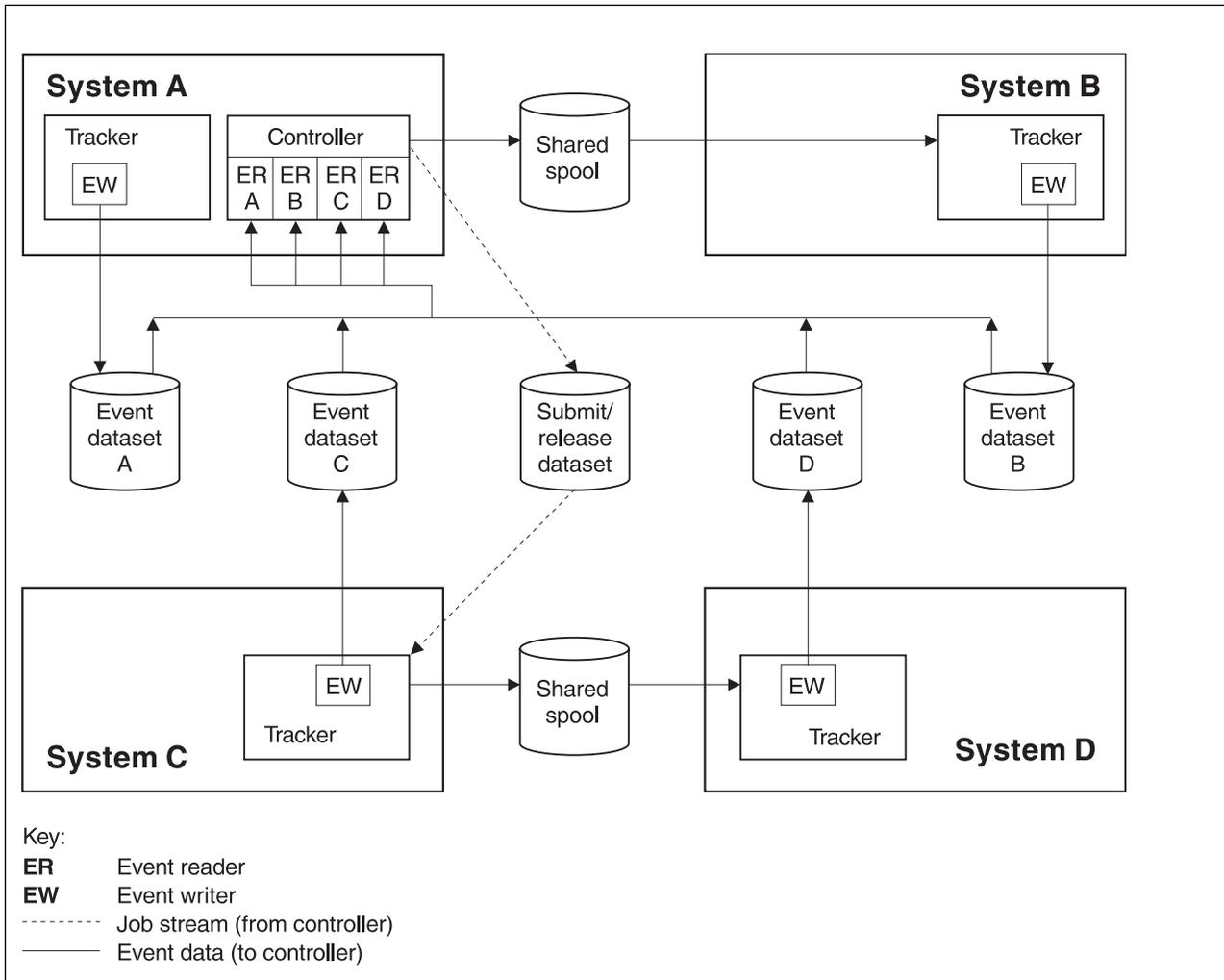


Figure 28. Two z/OS JES2 MAS complexes connected through shared DASD

Systems C and D, which are both IBM Workload Scheduler for z/OS controlled systems, form a second MAS complex. Work is sent to this complex via a submit/release data set. The destination field in the workstation description that represents this complex contains the DD name of the submit/release data set. The tracker on System C reads this data set and passes any new work to the complex for processing.

A tracker is installed on each system in the configuration. The event writer subtask of the tracker on each system writes events to an event data set on that system. Four event-reader subtasks, one for each of the event data sets, are started in the controller on System A. The controller reads the event data sets and updates the current plan.

When the controller is started on system A, it attempts to open the submit/release data set. If an I/O error occurs, the status of the workstation that represents the controlled MAS complex is set to offline. IBM Workload Scheduler for z/OS can then take automatic-workload-restart actions for operations at this workstation. These actions depend on the values that you specified on the WSOFFLINE keyword of the JTOPTS initialization statement.

Table 56 shows the initialization statements you can use to create the configuration in Figure 28 on page 372. This example assumes that some of the planned IBM Workload Scheduler for z/OS work on System C is submitted by a non-IBM Workload Scheduler for z/OS process. To control this work, the hold/release function is used. HOLDJOB(USER) is specified on the EWTROPTS statement for the tracker on System C. The RELDDNAME keyword is specified on the ERDROPTS statement of the event reader that reads event data set C. This keyword identifies the DD name of the submit/release data set that the controller should write release commands to.

Table 56. Example EQQPARM members for the previous figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES) ERDRTASK4(4) ERDRPARM(ERDRA,ERDRB) ERDRC,ERDRD	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKAEW)
ROUTOPTS DASD(SUDSC)	TRROPTS HOSTCON(DASD)
ERDRA	TRKAEW
ERDROPTS ERSEQNO(1)	EWTROPTS
ERDRB	
ERDROPTS ERSEQNO(2)	
ERDRC	
ERDROPTS ERSEQNO(3) RELDDNAME(SUDSC)	
ERDRD	
ERDROPTS ERSEQNO(4)	
EQQPARM members for System B	
TRACKERB	TRKBEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKBEW)	EWTROPTS
TRROPTS HOSTCON(DASD)	
EQQPARM members for System C	
TRACKERC	TRKCEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKCEW)	EWTROPTS SUREL(YES) HOLDJOB(USER)
TRROPTS HOSTCON(DASD)	
EQQPARM members for System D	

Table 56. Example EQQPARM members for the previous figure (continued)

EQQPARM members for System A		
TRACKERD		TRKDEW
OPCOPTS	OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKDEW)	EWTROPTS
TRROPTS	HOSTCON(DASD)	

Note: In this example, SUDSC is used for the user-defined DD name of the submit/release data set. This DD name appears in the started-task JCL of the controller, and in the destination field of the workstation that represents the controlled MAS system.

Individual systems connected via shared DASD

Figure 29 on page 375 shows three z/OS systems connected via shared DASD.

System A is the IBM Workload Scheduler for z/OS controlling system. Systems B and C are controlled systems, each of which shares a submit/release data set with the controlling system. Each of the three systems is represented by a computer workstation. The destination field in the workstation description for System A is left blank, indicating that IBM Workload Scheduler for z/OS should submit work to the system that the controller is started on. The destination field in the workstation descriptions for Systems B and C contains the DD name of the submit/release data set connecting them to the controller. Work is sent to the correct submit/release data set and is then passed to the corresponding system for processing by the event writer on that system.

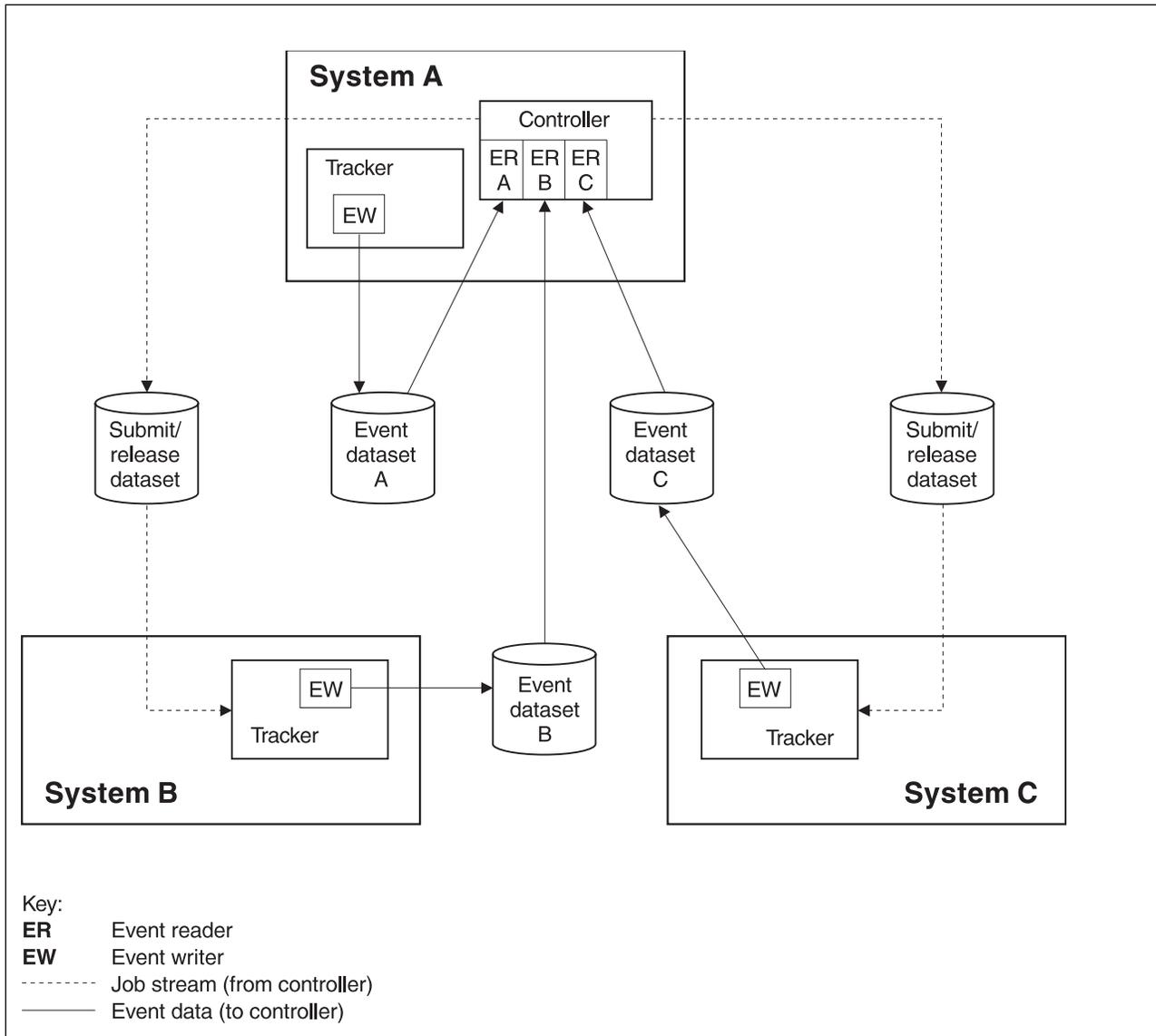


Figure 29. Individual systems connected through shared DASD

The event writer subtask on each system writes event information to its event data set. Three event-reader subtasks, one for each of the event data sets, are started in the controller on System A. The controller reads the event data sets and updates the current plan.

Automatic workload restart can be invoked in this configuration if an I/O error occurs when the controller attempts to open a submit/release data set. The workstation that has this submit/release data set as a destination is given the offline status, and WLR actions are taken according to the options specified on the WSOFFLINE keyword of the JTOPTS initialization statement.

Table 57 on page 376 shows the initialization statements you can use to create the configuration in Figure 29.

Table 57. Example EQQPARM members for the previous figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES) ERDRTASK(3) ERDRPARM(ERDRA,ERDRB, ERDRC)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKAEW)
ROUTOPTS DASD(SUDSB,SUDSC)	TRROPTS HOSTCON(DASD)
ERDRA	TRKAEW
ERDROPTS ERSEQNO(1)	EWTROPTS
ERDRB	
ERDROPTS ERSEQNO(2)	
ERDRC	
ERDROPTS ERSEQNO(3)	
EQQPARM members for System B	
TRACKERB	TRKBEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKBEW)	EWTROPTS SUREL(YES)
TRROPTS HOSTCON(DASD)	
EQQPARM members for System C	
TRACKERC	TRKCEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKCEW)	EWTROPTS SUREL(YES)
TRROPTS HOSTCON(DASD)	

Note: In this example, SUDSB and SUDSC are used for the user-defined DD names of the submit/release data sets. Both of these DD names appear in the JCL procedure of the controller. They also appear in the destination field of the respective workstations.

A z/OS Sysplex

“A z/OS Sysplex” shows four systems, each connected by cross-system coupling facility (XCF) communication links.

System A is the controlling IBM Workload Scheduler for z/OS system and Systems B, C, and D are controlled systems. You represent each system in the systems complex (Sysplex) by a computer workstation. The destination field contains the XCF-group-member name of the IBM Workload Scheduler for z/OS started task. On the controlling system, you can leave the destination field of the workstation that represents System A blank, or you can specify the XCF-group-member name of the tracker on that system. If you leave the field blank, the controller passes

work to the system for processing. If you specify the tracker XCF-group-member name, the controller transmits work to the tracker, which in turn passes the work to this system. The way that you define this workstation depends on the recovery strategy you want to use.

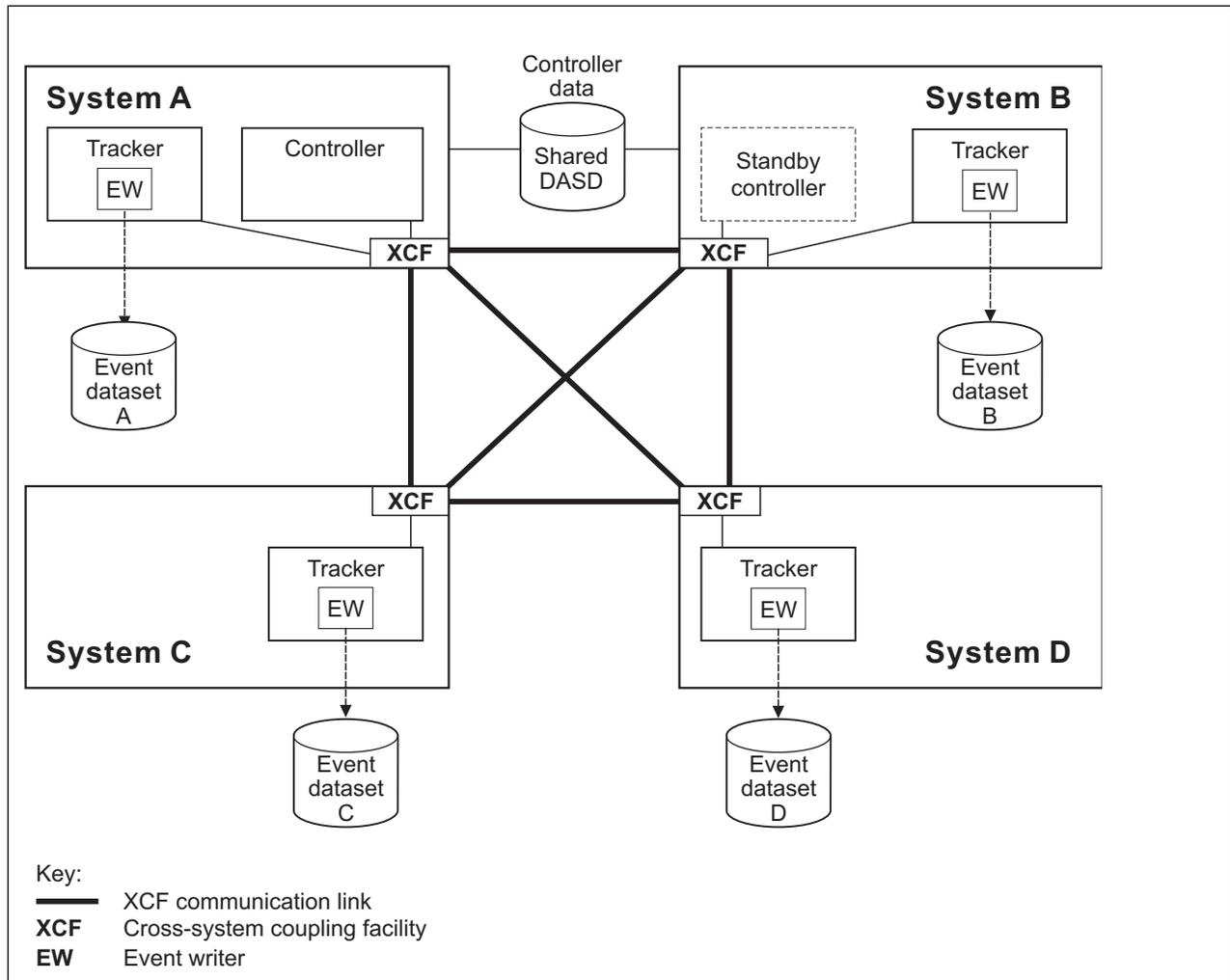


Figure 30. A z/OS Sysplex

A tracker is installed on each system in the sysplex. Each tracker event-writer subtask is started with a reader function, EWSEQNO is defined in the EWTROPTS statement. This means that the event writer passes the events to XCF for transfer to the controller at the same time as they are written to the event data set. This eliminates the need for separate event-reader subtasks.

XCF services let you define standby controllers, which act as a backup to the controller in case a failure occurs on the controlling system. This support is referred to as the hot standby function. In "A z/OS Sysplex" on page 376, an IBM Workload Scheduler for z/OS address space is started on System B in standby mode. It is a copy of the controller but does not perform any functions unless the controller fails or System A fails. The standby controller must have access to all IBM Workload Scheduler for z/OS data, because it becomes the controller in the event of a failure.

The full functions of workload restart are available in this configuration. If a z/OS system failure occurs, the workstation that represents that destination is set to failed. Actions are taken according to the WSFAILURE keyword of the JTOPTS initialization statement. If a tracker fails or if the communication link between the controller and the tracker fails, the workstation is set to offline. IBM Workload Scheduler for z/OS takes actions according to the WSOFFLINE keyword of JTOPTS.

Table 58 shows the initialization statements you can use to create the configuration in "A z/OS Sysplex" on page 376.

Table 58. Example EQQPARM members for the previous figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES) ERDRTASK(0)	OPCOPTS OPCHOST(NO) ERDRTASK(0)
ROUTOPTS XCF(SYSATRK,SYSBTRK, SYSCTRK,SYSDTRK)	EWTRTASK(YES) EWTRPARM(TRKAEW)
XCFOPTS GROUP(OPCGRP) MEMBER(CONTR)	XCFOPTS GROUP(OPCGRP) MEMBER(SYSATRK)
	TRROPTS HOSTCON(XCF)
	TRKAEW
	EWTROPTS EWSEQNO(1)
EQQPARM members for System B	
TRACKERB	STBYCONT
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKBEW)	OPCOPTS OPCHOST(STANDBY) ERDRTASK(0)
XCFOPTS GROUP(OPCGRP) MEMBER(SYSBTRK)	ROUTOPTS XCF(SYSATRK,SYSBTRK, SYSCTRK,SYSDTRK)
TRROPTS HOSTCON(XCF)	XCFOPTS GROUP(OPCGRP) MEMBER(STBYCTRB)
TRKBEW	
EWTROPTS EWSEQNO(1)	
EQQPARM members for System C	
TRACKERC	TRKCEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKCEW)	EWTROPTS EWSEQNO(1)
XCFOPTS GROUP(OPCGRP) MEMBER(SYSCTRK)	
TRROPTS HOSTCON(XCF)	
EQQPARM members for System D	

Table 58. Example EQQPARM members for the previous figure (continued)

EQQPARM members for System A	
TRACKERD	TRKDEW
OPCOPTS OPCHOST(NO)	EWTROPTS EWSEQNO(1)
ERDRTASK(0)	
EWTRTASK(YES)	
EWTRPARM(TRKDEW)	
XCFOPTS GROUP(OPCGRP)	
MEMBER(SYSDTRK)	
TRROPTS HOSTCON(XCF)	

Note: In this example, the XCF group is called OPCGRP. This group has members CONTR, SYSATRK, SYSBTRK, SYSCTRK, SYSDTRK, and STBYCTRB.

A PLEX configuration

Figure 31 on page 380 shows four systems running in a sysplex environment, connected using cross-system coupling facility (XCF) communication links.

One controller and one tracker are started on each tracker image of the sysplex; one controller becomes the active one, while the others start as standby controllers. One server is started on the z/OS image where the active controller runs, to handle requests from dialogs and PIF applications.

The &SYSCONE system variable is assumed to be set to KA, KB, KB, and KC on systems A, B, C, and D respectively.

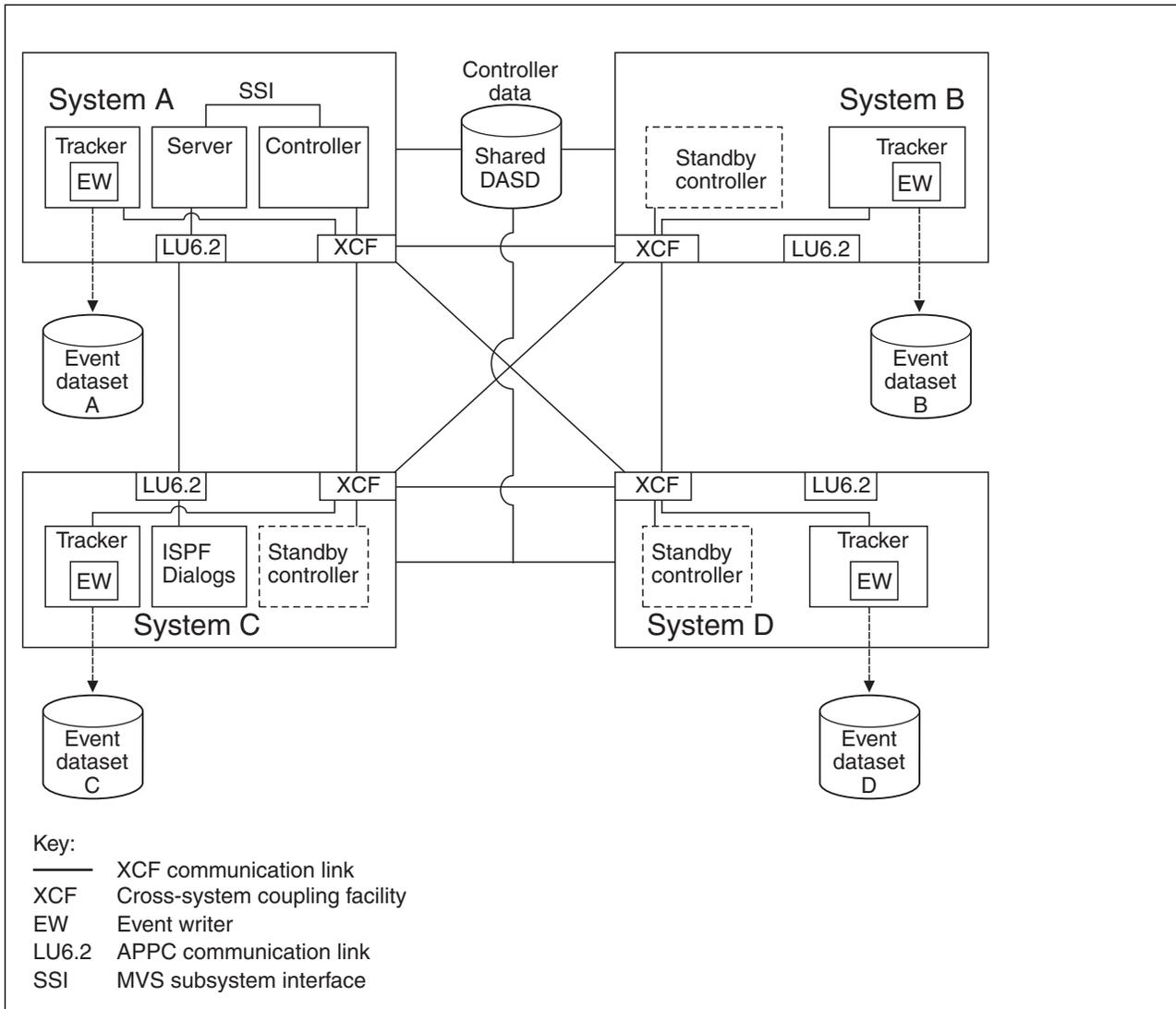


Figure 31. An IBM Workload Scheduler for z/OS PLEX environment

Table 59 shows the initialization statements you can use to create the configuration in Figure 31.

Table 59. Example EQQPARM Members for the previous figure

EQQPARM members, shared among z/OS images	
CONTROLR	SERVER
OPCOPTS	OPCHOST(PLEX)
	ERDRTASK(0)
	SERVERS(OSRV)
ROUTOPTS	XCF(TRKA, TRKB,
	TRKC, TRKD)
XCFOPTS	GROUP(OPCGRP)
	MEMBER(CONTR)
	SERVOPTS
	SUBSYS(OPCC)
	SCHEDULER(OSRV)

Table 59. Example EQQPARM Members for the previous figure (continued)

EQQPARM members, shared among z/OS images		
TRACKER		TRKEW
OPCOPTS	OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARAM(TRKB EW)	EWTROPTS EWSEQNO(1)
XCFOPTS	GROUP(OPCGRP) MEMBER(TR&SYSCLONE.)	
TRROPTS	HOSTCON(XCF)	

Controlling a z/OS system through a VTAM link

Figure 32 shows a z/OS system connected to the IBM Workload Scheduler for z/OS host via a VTAM link.

You represent each system by a computer workstation. The destination field in the workstation description for System A is left blank. Work for this workstation is started on System A. The destination field for the System B workstation contains the VTAM application ID of the tracker at this node. Work is transmitted from the host to the tracker and is then initiated on System B.

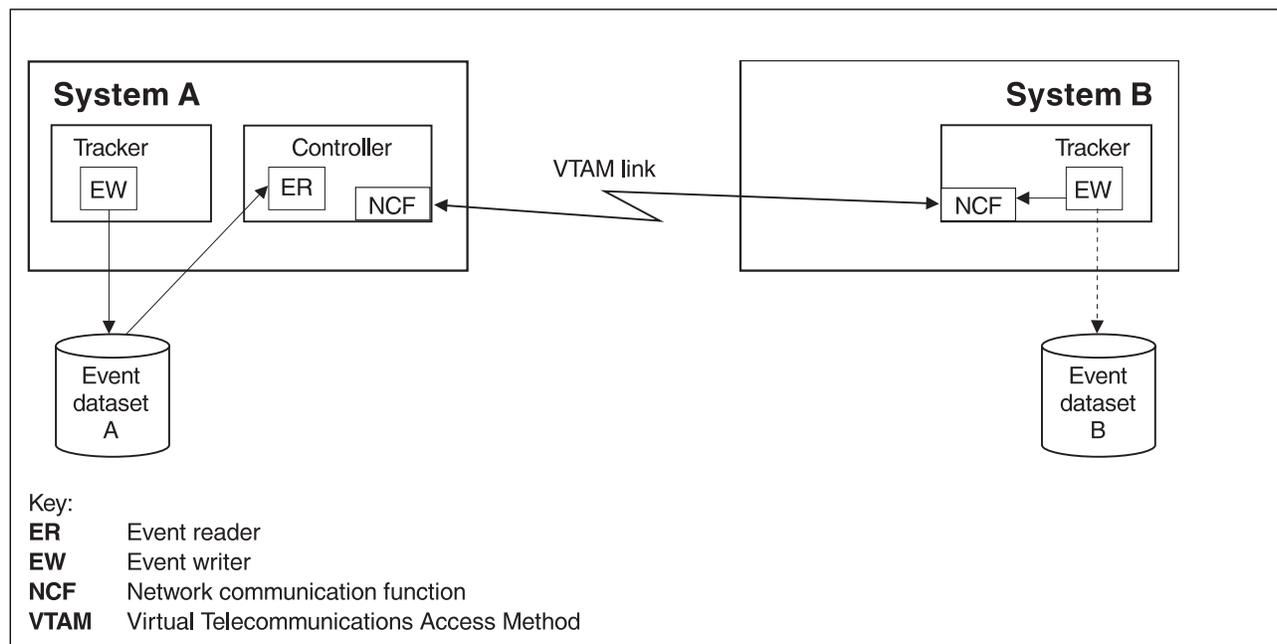


Figure 32. Controlling a z/OS system through a VTAM link

On System A, an event writer writes events to event data set A, which is read by an event reader subtask at the controller. On system B the tracker event-writer subtask is started with a reader function, EWSEQNO is defined in the EWTROPTS statement. This means that the event writer passes the events to NCF for transfer to the controller at the same time as they are written to the event data set.

Automatic workload restart can be used in this configuration if the controller cannot communicate with the tracker on system B. The status of the workstation for System B is set to offline if z/OS is stopped or fails, if the tracker is stopped or

fails, or if the VTAM link is lost. WLR actions are taken according to the WSOFFLINE keyword of the JTOPTS initialization statement.

Table 60 shows the initialization statements you can use to create the configuration in Figure 32 on page 381.

Table 60. Example EQQPARM Members for the previous figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES) ERDRTASK(1) ERDRPARM(ERDR1) NCFTASK(YES) NCFAPPL(NCFAPPL1)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKAEW)
ROUTOPTS SNA(NCFAPPL2)	TRROPTS HOSTCON(DASD)
ERDR1	TRKAEW
ERDROPTS ERSEQNO(1)	EWTROPTS
EQQPARM members for System B	
TRACKERB	TRKBEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKBEW) NCFTASK(YES) NCFAPPL(NCFAPPL2)	EWTROPTS EWSEQNO(1)
TRROPTS HOSTCON(SNA) SNAHOST(NCFAPPL1)	

Note: In this example, the controller has VTAM application ID NCFAPPL1, and the tracker on System B has VTAM application ID NCFAPPL2.

Controlling a z/OS system through a TCP/IP link

Figure 33 on page 383 shows a z/OS system connected to the IBM Workload Scheduler for z/OS host via a TCP/IP link.

You represent each system by a computer workstation. The destination field in the workstation description for System A is left blank. Work for this workstation is started on System A. The destination field for the System B workstation contains the destination name associated with the IP address of the tracker on this system. Work is transmitted from the host to the tracker and is then initiated on System B.

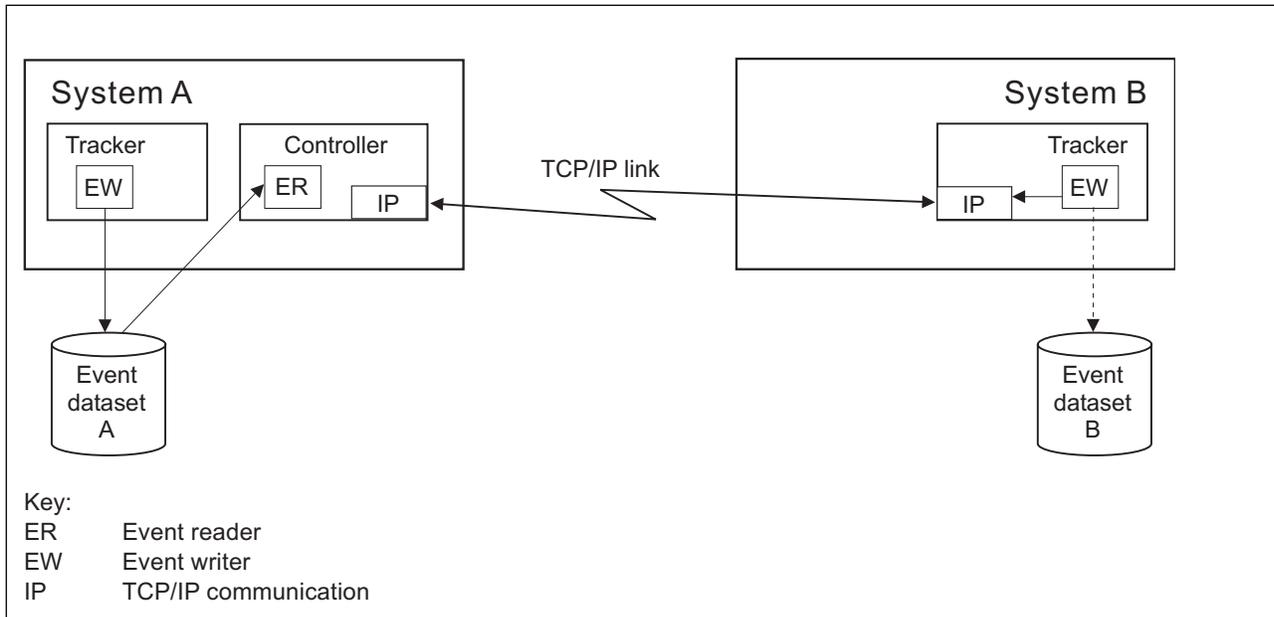


Figure 33. Controlling a z/OS system through a TCP/IP link

On System A, an event writer writes events to event data set A, which is read by an event reader subtask at the controller. On system B the tracker event-writer subtask is started with a reader function, EWSEQNO is defined in the EWTROPTS statement. This means that the event writer passes the events to NCF for transfer to the controller at the same time as they are written to the event data set.

Automatic workload restart can be used in this configuration if the controller cannot communicate with the tracker on system B. The status of the workstation for System B is set to offline if z/OS is stopped or fails, if the tracker is stopped or fails, or if the link is lost. WLR actions are taken according to the WSOFFLINE keyword of the JTOPTS initialization statement.

Table 61 shows the initialization statements you can use to create the configuration in Figure 33.

Table 61. Example EQQPARM Members for the previous figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES)	OPCOPTS OPCHOST(NO)
ERDRTASK(1)	ERDRTASK(0)
ERDRPARM(ERDR1)	TRROPTS HOSTCON(DASD)
TCPOPTS TCPIPJOBNAME('TCPIP')	
HOSTNAME('9.12.134.1')	
TRKPORTNUMBER(8888)	
ROUTOPTS TCPIP(DEST1:'1.111.111.111'/4444)	
ERDR1	TRKAEW
ERDROPTS ERSEQNO(1)	EWTROPTS
EQQPARM members for System B	

Table 61. Example EQQPARM Members for the previous figure (continued)

EQQPARM members for System A	
TRACKERB	TRKBEW
OPCOPTS	OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARAM(TRKBEW)
TCPOPTS	TCPIPJOBNAME('TCPIP') HOSTNAME('1.111.111.111') TRKPORTNUMBER(4444)
TRROPTS	HOSTCON(TCP) TCPHOSTNAME('9.12.134.1') TCPPORTNUMBER(8888)
	EWTROPTS EWSEQNO(1)

Note: In this example, the name of the destination is DEST1. The destination is defined also in the destination field of the workstation.

Controlling a JES2 MAS system through a VTAM link

Figure 34 on page 385 shows a z/OS JES2 MAS system connected to the IBM Workload Scheduler for z/OS host via a VTAM link.

System A and the systems in the JES2 MAS complex (System B and System C) are each represented by a computer workstation. The destination field for the workstation on System A is left blank so that work is initiated by the controller on that system. The destination field of the workstation descriptions for the MAS complex contains the VTAM application ID of the tracker on System B. The controller sends work to the tracker on System B via the network communication function. The tracker passes the work to the complex, and the work then processes on either System B or System C, depending on installation parameters.

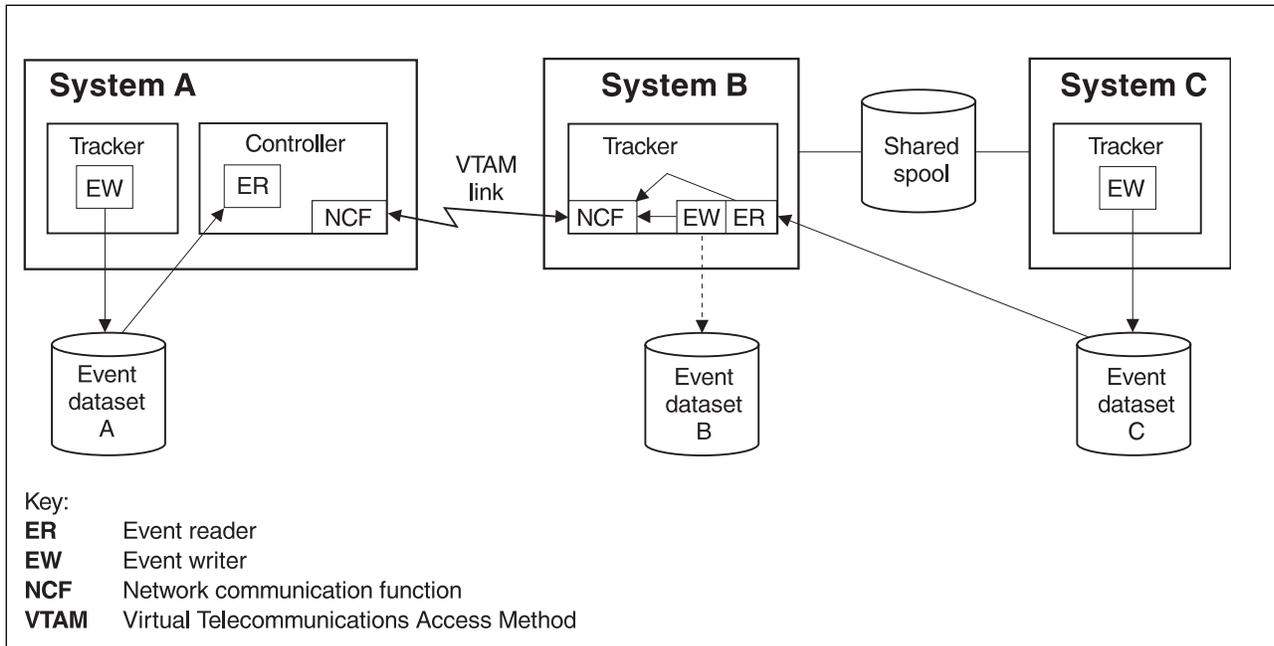


Figure 34. Controlling a JES2 MAS system through a VTAM link

A tracker is started on each system in the configuration. An event-reader subtask in the controller reads events from System A. The event-reader on System B reads the event information from System C and passes the events to NCF for transmission to the controller. This event-reader is required because System C does not have its own link to the controller. The event-writer subtask on System B is started with a reader function—EWSEQNO is defined in the EWTROPTS statement. This means that the event writer passes the events for System B to NCF for transfer to the controller at the same time as they are written to the event data set.

Note: This figure demonstrates the need for an event reader task where System C does not have a direct link to the controller. But if the required resources are available, try to give each tracker its own link to the controller.

Automatic workload restart can be used in this configuration if the controller cannot communicate with the tracker on System B. The status of the workstation for System B is set to offline if z/OS is stopped or fails, if the tracker is stopped or fails, or if the VTAM link is lost. WLR actions are taken according to the WSOFFLINE keyword of JTOPTS. Workload restart is not affected by failures on System C, because the controller has no direct link with this system.

Table 62 on page 386 shows the initialization statements you can use to create the configuration in Figure 34.

Table 62. Example EQQPARM members for the preceding figure

EQQPARM members for System A	
CONTROLR	TRACKERA
OPCOPTS OPCHOST(YES) ERDRTASK(1) ERDRPARM(ERDR1) NCFTASK(YES) NCFAPPL(NCFAPPL1)	OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKAEW)
ROUTOPTS SNA(NCFAPPL2)	TRROPTS HOSTCON(DASD)
ERDR1	TRKAEW
ERDROPTS ERSEQNO(1)	EWTROPTS
EQQPARM members for System B	
TRACKERB	TRKBEW
OPCOPTS OPCHOST(NO) ERDRTASK(1) ERDRPARM(ERDR2) EWTRTASK(YES) EWTRPARM(TRKBEW) NCFTASK(YES) NCFAPPL(NCFAPPL2)	EWTROPTS EWSEQNO(1)
TRROPTS HOSTCON(SNA) SNAHOST(NCFAPPL1)	
ERDR2	
ERDROPTS ERSEQNO(2)	
EQQPARM members for System C	
TRACKERC	TRKCEW
OPCOPTS OPCHOST(NO) ERDRTASK(0) EWTRTASK(YES) EWTRPARM(TRKCEW)	EWTROPTS HOLDJOB(NO)
TRROPTS HOSTCON(DASD)	

Note: In this example, the controller has VTAM application ID NCFAPPL1, and the tracker on System B has VTAM application ID NCFAPPL2.

Appendix C. Invoking the EQQEXIT macro

The sample event-tracking exits shipped with IBM Workload Scheduler for z/OS are written in assembler language. The event-tracking code in these exits is generated by an assembler macro called EQQEXIT. The following sections describe how you invoke the EQQEXIT macro. This appendix contains General-use Programming Interface and Associated Guidance Information.

Invoking EQQEXIT in SMF exits

EQQEXIT establishes its own addressability in SMF exits. It saves and restores all used registers. To do this, it expects Register 13 to point to a standard z/OS save area.

There are two ways to invoke the EQQEXIT macro in an SMF exit:

- Invoke EQQEXIT with all registers unchanged since the exit was called (except Register 15).
- Save all registers on entry to the exit and then invoke EQQEXIT by specifying the address of the initial save area.

In both cases, the EQQEXIT macro must be invoked in Supervisor state, PSW key 0.

Invoking EQQEXIT in JES exits

In JES exits, EQQEXIT must be invoked in Supervisor state, PSW key 1. EQQEXIT expects code addressability to be already established. It also expects registers to be set up as follows:

- **EXIT7**
 - R0** JCT read/write indicator (JES2 SP Version 3, or earlier); address of a parameter list mapped by the JES2 \$XPL macro (JES2 SP Version 4, or later).
 - R1** Address of the JCT being read or written.
 - R13** Address of the current PCE.
- **EXIT51**
 - R1** Address of a parameter list mapped by the JES2 \$XPL macro (JES2 with z/OS 1.7, or later).
- **IATUX19**
 - R8** Address of the current JDS entry.
 - R9** Address of the current RESQUEUE entry.
 - R11** Address of the current FCT entry.
 - R12** Address of the TVTABLE entry.
- **IATUX29**
 - R11** Address of the current FCT entry.
 - R13** Address of the input-service data area for the current function.

Note that these register conventions are already set up when the exit is called. You must invoke EQQEXIT while these registers are unchanged.

If a shipped JES exit example (or the EQQEXIT macro) has been user-modified, make sure that it does not prevent or filter the tracking of IBM Workload Scheduler for z/OS itself.

See the NOTES section of the EQQEXIT prolog for information about the register contents that are destroyed by EQQEXIT in JES exits.

Macro invocation syntax for EQQEXIT

Purpose

EQQEXIT produces IBM Workload Scheduler for z/OS event-tracking exit code by generating assembler code to perform in an SMF or JES exit.

Syntax

EXIT=*exit name*
REG13=*address of save area*
MAPMAC={YES | NO}
SETUID={YES | NO}
SRREAD={YES | NO | NONE}
SNA={YES | NO}

Parameters

EXIT=*exit name*

A required keyword defining the name of the exit in which the macro is used. The following names can be specified: IEFACRT, IEFUJI, IEFU83, EXIT7, IATUX19, and IATUX29. Except for the EXIT7 exit, a warning message is issued if the name of the current CSECT differs from the name specified by the EXIT keyword.

REG13=*address of save area*

An optional keyword defining the address of the current-register save area when the SMF or JES exit was called. The default for this keyword depends on the name specified by the EXIT keyword. If the current exit is EXIT7, the default is PCELPSV. If the current exit is IATUX19 or IATUX29, the default is FCTSAVCH. In all other cases, the default is the second fullword in the current save area (if the current save area is properly chained, and the previous save area contains the registers at entry to the exit).

If the default does not apply, the REG13 keyword must be specified. Its value must be a fullword pointing to the save area that was used to store all the registers when the exit was entered.

MAPMAC={YES | NO}

An optional keyword specifying whether the macro should generate the required assembler mapping macros. The default is to generate these mapping macros. The following mapping macros are required by EQQEXIT code: CVT, IEFJESCT, IEFJSSOB, and IEFJSSIB. The IEFACRT exit also requires the IEFJMR macro.

If you specify NO, the IEFU83 exit requires mapping of the SMF records IFASMFR 14 and IFASMFR 64. You must label them SMF14REC and SMF64REC, respectively. For example:

```
SMF14REC DSECT          * SMF RECORD 14 MAPPING
IFASMFR 14             * DATA SET ACTIVITY RECORD
```

SETUID={YES|NO}

An optional keyword specifying whether the macro should generate code to place the current user ID in the JMRUSEID field when the IEFUJI exit is taken. Specify YES to generate this code. If you specify NO, which is the default, the JMRUSEID field is not updated. You are recommended to specify YES if you use the current user ID to filter data set close events. You need these mapping macros when you specify YES: IHAPSA, IHAASCB, IHAASXB, and IHAACEE.

SRREAD={YES|NO|NONE}

An optional keyword defining whether a resource availability event should be generated when a data set is closed after being opened for read processing.

When YES is specified, an SR event is generated each time a data set is closed after being opened for either read or output processing.

When NO is specified or defaulted, the SR event is generated only when a data set has been opened for output processing. The event is not generated if the data set has been opened for read processing.

When you specify NONE, no data set triggering is performed.

See “Implementing support for data set triggering” on page 92 for more information about the data set triggering function.

SNA={YES|NO}

An optional keyword specifying whether JES3 SNA NJE is supported.

Messages

The following messages can be generated at assembly time:

- WARNING: SNA KEYWORD IS ONLY USED FOR EXIT = IATUX19
- WARNING: SNA VALUE SNA IS NOT RECOGNIZED
- WARNING: EXIT NAME DIFFERS FROM CURRENT CSECT NAME
- WARNING: MAPMAC VALUE MAPMAC IS NOT RECOGNIZED
- WARNING: SRREAD KEYWORD IS ONLY USED FOR EXIT=IEFU83
- WARNING: SRREAD VALUE NOT RECOGNIZED, YES OR NO ARE THE ONLY VALID VALUES
- EXIT NAME *EXIT* IS NOT SUPPORTED

Return codes

The following return codes can be generated at assembly time:

- 4 Input invalid, check for warning messages.
- 12 Unsupported exit specified for the EXIT keyword.

Appendix D. Invoking the EQQSENT macro

The following procedure is supported only for compatibility with earlier versions. To use the current support for data set triggering, see the procedure for running event-driven workload automation described in *Managing the Workload*.

When the data set triggering function is used, you specify the data sets for which you want events generated by building the data set selection table EQQDSLST. The EQQDSLST is created by invoking the EQQSENT macro. The following sections describe how you invoke the EQQSENT macro. This appendix contains General-use Programming Interface and Associated Guidance Information.

Note: The current support for data set triggering is based on the EQQEVLSST configuration file. If EQQJCLIB contains both EQQEVLSST and EQQDSLST, the resulting triggering selection table is the union of EQQEVLSST and EQQDSLST. In this case, EQQEVLSST data is processed first. If EQQJCLIB contains only EQQDSLST, the tracker loads it as triggering selection table.

Invoking EQQSENT to create EQQDSLST

The EQQSENT macro is used to create entries in the data set triggering selection table. The selection table is loaded into ECSA when the IBM Workload Scheduler for z/OS event writer is started.

The sample EQQLSJCL in the SEQQSAMP library can be used to invoke the EQQSENT macro.

Macro invocation syntax for EQQSENT

Purpose

EQQSENT produces an entry in the data set triggering selection table, EQQDSLST. EQQDSLST is used in SMF exit IEFU83 by the data set triggering function to decide which SMF records to process. When an SMF 14, 15, or 64 record matches a condition in EQQDSLST, a special resource availability event is created and broadcast to all IBM Workload Scheduler for z/OS subsystems defined on the system where the SMF record was created.

Format

STRING= *string* | **LASTENTRY**
POS= *numeric position*
USERID= *user ID filter criteria*
JOBNAME= *jobname filter criteria*
AINDIC={Y|N}
LIFACT={Y|N|R}
LIFTIM=*interval*

Parameters

STRING=*string* | **LASTENTRY**

Required keyword specifying the character string to be searched for. The string can be 1 to 44 characters long. To identify the fully-qualified last

level of a data set name, add a space as the last character and enclose the string in single quotation marks. Consider this example. You have two data sets:

```
DSN.NAME.AB
DSN.NAME.ABC
```

Specify `STRING=DSN.NAME.AB,POS=1` if you want SR availability events created for both data sets. Specify `STRING='DSN.NAME.AB ',POS=1` if you want events created only for the first data set.

When `EQQLSENT` is invoked with `STRING=LASTENTRY` it generates an end of table indicator. After having invoked `EQQLSENT` with keyword parameters `STRING` and `POS` a number of times, `EQQLSENT` must be invoked one last time with `STRING=LASTENTRY` in order to complete the table.

To create an empty `EQQDSLST`, just invoke `EQQLSENT` once, with `STRING=LASTENTRY`. When an empty list is used by `IEFU83`, no SR events are created.

POS=numeric position

A required keyword specifying the numeric position where the string begins.

USERID=string

Optional keyword specifying a generic character string to be compared with the `SMFxxUID` field, which contains the user identification associated with the job, started task, or TSO user that requested the activity against the data set that resulted in the data set close. The string can be 1 to 8 characters long. For this parameter the following wildcards are allowed:

- *: to match any sequence of characters.
- %: to match any single character. For example, if you specify `AB%`, `ABC` is a match, `AB` or `ABCD` are not a match.

Note: The SMF user ID field may contain a blank value. See *z/OS System Management Facilities* for more information about the `SMFxxUID` or `SMFxxUIF` field.

If you need to control SR availability events based on the user ID and the SMF value is blank in your installation, consider using the `IEFUJI` exit to insert the user ID. You are recommended to specify `SETUID=YES` on the `EQQEXIT` macro when you generate the `IEFUJI` exit: this sets the `JMRUSEID` field, which SMF then copies to the SMF user ID field.

If you want to update the `JMRUSEID` field yourself, the user ID is most easily taken from the `ACEEUSRI` field in the `ACEE`, pointed to from the `ASXB`, pointed to from the `ASCB`. This can be located as follows:

```
PSAAOLD ==> ASCB          ACSBASXB ==>
ASXB                      ASXBSENV ==> ACEE
                           ACEEUSRI ==> userid
```

The DSECTs needed are mapped by these macros:

Area	Macro	Library
PSA	IHAPSA	SYS1.MACLIB
ASCB	IHAASCB	SYS1.MACLIB
ASXB	IHAASXB	SYS1.MODGEN

Area	Macro	Library
ACEE	IHAACEE	SYS1.MACLIB

The JMR, mapped by IEFJMR, is already available in the EQQEXIT expansion in IEFUJI.

JOBNAME=string

Optional keyword specifying a generic character string to be compared with the SMF14JBN, SMF15JBN, or SMF64JMN field, which contains the name of the job, started task or TSO user that requested the activity against the data set that resulted in the data set close. The string can be 1 to 8 characters long. For this parameter the following wildcards are allowed:

- *: to match any sequence of characters.
- %: to match any single character. For example, if you specify AB%, ABC is a match, AB or ABCD are not a match.

If the data set is to be processed by FTP, JOBNAME corresponds to the **USERID** under which the data set is received. That is, the USERID supplied when the remote host opened the FTP session to PUT the data set, or when a local user (or batch job) opened the FTP session to GET the data set.

AINDIC={Y|N}

Optional keyword specifying that the special resource is available (Y) or unavailable (N). The default is that the resource available.

LIFACT={Y|N|R}

Optional keyword specifying the value to which the global availability of the special resource is reset, after the interval of time specified by LIFTIM has expired. Allowed values are:

- Y** Sets the global availability to Yes
- N** Sets the global availability to No
- R** Sets the global availability to blank

This keyword is valid only if LIFTIM is specified. The default is R.

LIFTIM=interval

Optional keyword specifying the interval of time, in minutes, after which the global availability of the special resource is reset to the value specified by LIFACT. The allowed range is from 1 to 999999.

Note:

1. The output from assembling the EQQLSENT macro must be placed in the EQQDSLST member in the data set referenced by the ddname EQQJCLIB.
2. Generation Data Group data sets are specified by the group name. For example, when a GDG data set with the name 'DSN.OPCSUBS.GDG.G0001V00' is closed the special resource event contains resource name 'DSN.OPCSUBS.GDG'.
3. For a partitioned data set, the member name is not part of the resource name in the SR event.
4. For VSAM data sets the resource name in the SR event is the cluster name (without the DATA or INDEX suffix).

Examples

```
EQQLSENT STRING=SYS1.MAN,POS=1
EQQLSENT STRING='TEST.DSCLOSE ',POS=1,USERID=SYSOP
EQQLSENT STRING=CP2,POS=12
EQQLSENT STRING=EQQDATA.EXCL,POS=5
EQQLSENT STRING='DSN.OPCSUBS.GDG ',POS=1
EQQLSENT STRING=LASTENTRY
END
```

In this example, SMF records with:

- A data set name beginning with SYS1.MAN, or
- Data set name TEST.DSCLOSE and user ID SYSOP
- Records with CP2 in position 12, such as DSN.OPCSUB.CP2, or
- Records that have EQQDATA.EXCL starting in position 5
- The root of a GDG data set name

will cause SR availability events to be generated.

Messages

The following messages can be generated at assembly time:

- KEYWORD STRING IS REQUIRED
- KEYWORD POS IS REQUIRED
- POSITION MUST BE BETWEEN 1 AND 43
- NULL NAME NOT VALID
- NAME (STRING) GREATER THAN 44 CHARACTERS
- POSITION INVALID FOR NAME (STRING)
- USERID STRING NOT VALID
- JOBNAME STRING NOT VALID
- AINDIC MUST BE EITHER Y OR N
- POSITION NOT VALID FOR NAME (STRING)
- LIFACT MUST BE Y, N, OR R
- LIFTIM LENGTH NOT VALID
- LIFTIM VALUE NOT VALID
- LIFTIM VALUE 0 NOT ALLOWED

Return codes

The following return code can be generated at assembly time:

12 Input invalid, check error messages.

Appendix E. Using response files

You can silently install the Dynamic Workload Console, using a response file. For more information about how to silently install, see “Performing a silent installation” on page 264. The response file properties are described in “The Dynamic Workload Console response file properties.”

The Dynamic Workload Console response file properties

The following tables describe the properties used in the Dynamic Workload Console response file:

- General information: Table 63.
- WebSphere configuration: Table 64 on page 396.
- z/OS connector configuration: Table 65 on page 397.
- Offering and features to install: Table 66 on page 398.
- Upgrade settings: Table 67 on page 398.

Note:

1. All values must be written between single quotation marks ('), for example:
<data key='user.wasInstallLocation,com.ibm.tws' value='C:\Program Files\IBM\TWA\' />
2. Properties are written in mixed case for ease of reading, but are not case-sensitive.
3. Keywords (for example, "true") used in values, are not case-sensitive.
4. For UNIX and Linux operating systems, in case the installation is performed by a non-root user, customize the installation path properly.

Table 63. General information

Name	Description	Permitted values
repository location	The location where you stored the eImages.	For more information about downloading eImages on your workstation, see Downloading eImages on your workstation

Table 63. General information (continued)

Name	Description	Permitted values
profile id	The profile ID.	<p>Fresh installation The value must be <i>IBM Dynamic Workload Console</i>. Do not modify this value.</p> <p>Upgrade The value depends on the original fresh installation of the product. If the original fresh installation of the product was version 9.2 Fix Pack 1 or later, then the value must be <i>IBM Dynamic Workload Console</i>. If the original fresh installation of the product was a version earlier than version 9.2 Fix Pack 1, then the value must be <i>Tivoli Dynamic Workload Console</i>.</p>
installLocation	The Dynamic Workload Console installation directory.	For more information about the installation directory possible values, see "Installation procedure for Dynamic Workload Console" on page 259.
user.offeringId	The offering ID.	The value must be <i>com.ibm.tws.tdwc</i> . Do not modify this value.

Table 64. WebSphere configuration

Name	Description	Permitted values
user.wasInstallLocation	The directory where you installed the WebSphere Application Server.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.
createNewWasProfile	To create a new WebSphere Application Server profile after the installation.	<p>false</p> <p>Do not modify the value in the response file.</p>
user.wasProfileLocation	The location where the WebSphere Application Server profile is saved.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.
user.wasUserName	Enter the WebSphere Application Server user ID of the WebSphere Application Server profile that you use. This field is optional.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.

Table 64. WebSphere configuration (continued)

Name	Description	Permitted values
user.wasPassword	Enter the encrypted password of the WebSphere Application Server user ID of the WebSphere Application Server profile that you are using. For more information about password encryption, see "Encrypting user passwords for response files" on page 268.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.

Table 65. z/OS connector configuration

Name	Description	Permitted values
user.zosConnIsEnabled	Specify if you want to create a connection to an IBM Workload Scheduler for z/OS host.	<p>true Installation process configures a new connection to an IBM Workload Scheduler for z/OS controller.</p> <p>false Installation process does not configure a connection to an IBM Workload Scheduler for z/OS controller. Note: After the installation, you can create connections using <i>wastools</i> scripts. The default is false.</p>
user.zosConnEngineName	Specify the name of the IBM Workload Scheduler for z/OS engine which you are connecting to.	It is a label that identifies the z/OS connector instance.
user.zosConnHostname	Specify the host name or TCP/IP address of the remote z/OS system where the IBM Workload Scheduler for z/OS controller is installed.	A valid host name or TCP/IP address.
user.zosConnPort	Specify the number of the TCP/IP port of the remote z/OS system used to communicate with the IBM Workload Schedulerz/OS controller.	This value must correspond to the value specified in the SERVOPTS member of the controller. The default value is 11111 .
user.zosConnSslIsEnabled	Specify if you want to create the connection to an IBM Workload Scheduler for z/OS controller in SSL mode.	<p>true Installation process configures the connection to an IBM Workload Scheduler for z/OS controller in SSL mode.</p> <p>false Installation process does not configure a connection to an IBM Workload Scheduler for z/OS controller in SSL mode. Note: After the installation, you can create connections using <i>wastools</i> scripts. The default is false.</p>

Table 66. Offering and features to install

Name	Description	Permitted values
modify	This property is used by silent installation as-is, and must not be modified. In this boolean field the installation process specifies if you are modifying the product already installed.	Do not modify the value in the response file.
offering id	This property is used by silent installation as-is, and must not be modified.	Do not modify the value in the response file.
profile	This property is used by silent installation as-is, and must not be modified.	Do not modify the value in the response file.
feature	This property is used by silent installation as-is, and must not be modified.	Do not modify the value in the response file.
installFixes	This property is used by silent installation as-is, and must not be modified. In this boolean field the installation process specifies if you are installing fixes to the product.	Do not modify the value in the response file.

Table 67. Upgrade settings

Name	Description	Permitted values
repository location	The location where you stored the eImages.	For more information about downloading eImages on your workstation, see Downloading eImages on your workstation
repository location	The location where you stored the fix pack eImages.	For more information about downloading eImages on your workstation, the readme files available on Fix Central.
profile id	The profile ID.	<p>Fresh installation</p> <p>The value must be <i>IBM Dynamic Workload Console</i>. Do not modify this value.</p> <p>Upgrade</p> <p>The value depends on the original fresh installation of the product. If the original fresh installation of the product was version 9.2 Fix Pack 1 or later, then the value must be <i>IBM Dynamic Workload Console</i>. If the original fresh installation of the product was a version earlier than version 9.2 Fix Pack 1, then the value must be <i>Tivoli Dynamic Workload Console</i>.</p>

Table 67. Upgrade settings (continued)

Name	Description	Permitted values
user.offeringId	The offering ID.	The value must be <i>com.ibm.tws.tdwc</i> . Do not modify this value.
user.wasUserName	Enter the WebSphere Application Server user ID of the WebSphere Application Server profile that you use. This field is optional.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.
user.wasPassword	Enter the encrypted password of the WebSphere Application Server user ID of the WebSphere Application Server profile that you are using. For more information about password encryption, see "Encrypting user passwords for response files" on page 268.	For more information about the configuration of the WebSphere Application Server profile, see "WebSphere Application Server profile configuration" on page 263.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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
US*

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*

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 jurisdictions 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.

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 provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises 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.

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and 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 on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL is a Registered Trade Mark of AXELOS Limited.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

Index

A

ACCEPT processing 363
accessibility xiv
activating
 API (application programming interface) 147
 Dynamic Workload Console 155
 NCF 143
 server 150
 subtasks 11
additional plug-ins
 uninstallation procedures 325
agent 335, 341
 maintaining 327
 on IBM i 335, 341
 troubleshooting 327
 with z-centric capabilities 315
agent dynamic
 on IBM i 335, 341
agent installation
 scanning system prerequisites 256
agent installation return code
 twinsinst 327, 349
agent restore return code
 twinsinst 327, 349
agent uninstallation return code
 twinsinst 327, 349
agent uninstalling
 twinsinst 345
agent upgrade
 scanning system prerequisites 256
agent upgrade return code
 twinsinst 327, 349
agent with z-centric capabilities
 installing 309
 dockerfile 315
allocating data sets
 non-VSAM 109
APAR
 PK93917 89
APARs
 IZ79105 277
 PI13017 88
 PI19400 134, 157
 PI63875 196
 PI67317 90, 186
 PI77416 103, 104
 PI81106 134
 PI97370 85
 PK00502 146
 PK04155 93, 95
 PK05336 200
 PK06007 81, 86
 PK06227 67, 81, 82, 164, 165, 355, 366, 387
 PK06712 145
 PK06763 194
 PK11767 364
 PK16903 137, 138
 PK18760 41
 PK20776 89

APARs (*continued*)
 PK20879 200
 PK22761 190
 PK23181 126
 PK25245 125
 PK25268 107
 PK25979 359
 PK28707 115
 PK30374 3
 PK31005 99
 PK34310 98, 125
 PK39432 192
 PK40356 67, 154, 355, 367
 PK40969 7, 9, 10, 18, 19, 26, 42, 44, 60, 62, 76, 99, 146, 171, 182
 PK41519 72
 PK45614 156
 PK53014 139
 PK53476 130, 132
 PK54782 200
 PK56520 98, 130, 133
 PK57062 139
 PK64650 393
 PK65147 366
 PK65527 137
 PK69493 93
 PK77418 49
 PK81179 391
 PK83161 200
 PK86682 57, 58, 69, 73
 PK88734 81
 PK92042 83, 145, 364, 365
 PK94896 5
 PK96348 107, 108, 208
 PM01090 110, 124
 PM02690 124
 PM04245 136, 140
 PM06648 109, 110, 194
 PM07439 67, 356, 357, 364
 PM08778 92, 115
 PM21607 136
 PM32308 126
 PM45677 89
 PM60019 109
 PQ65923 3
 PQ78043 3
 PQ81700 102
 PQ84095 364
 PQ86050 109
 PQ87576 93, 97
 PQ87710 126
 PQ89715 123
 PQ91074 129
 PQ96400 84
 PQ96540 41, 53, 72, 84, 119
 PQ97143 98
 PQ98852 127
 PQ99317 67, 358
 PQ99366 207
API (application programming interface)
 activating support for 91, 147

APPC/MVS
 options, updating in
 SYS1.PARMLIB 91, 149, 153
APPCPMnn member of
 SYS1.PARMLIB 91, 149, 153
APPL
 resource class 95
 statements
 API (application programming interface) 148
 local LU for the server,
 defining 152
 local LU, defining 148
 NCF 144, 146
application description data set
 (EQQADDS) 102
 considerations when allocating 106
application job plug-ins
 option to add runtime for Java
 runtime to run job types with
 advanced options 342
application node definitions, NCF 144
application programming interface (API)
 activating support for 91, 147
application server
 installation log files 272
APPLY processing 362
applying maintenance
 z/OS connector 241
 z/OS connector on WebSphere
 Application for z/OS 233
archive of current plan (EQQACPDS)
 considerations when allocating 106
AUTHCMD statement, updating for
 Tivoli OPC TSO commands 90
authority, problem determination
 procedures 170
authorization roles
 twinsinst 310
 z/OS connector on WebSphere
 Application for z/OS 233
AUTHTSF statement, updating for
 EQQMINOx 90
automatic-recovery-procedure library
 (EQQPRLIB) 110
 considerations when allocating 121
available functions
 for dynamic workload broker 273
 for IBM Workload Scheduler 273

B

backup checkpoint data set
 (EQQBKPT) 109
BACKUP command 90
backup controller
 migrating 192
batch jobs
 security 93
 user ID of Tivoli OPC submitted
 jobs 93

- batch-jobs
 - generating skeleton JCL 69
- BULKDISC command 90
- business scenario
 - z/OS connector on WebSphere
 - Application Server for z/OS 233

C

- calendar and workstation data set (EQQWSDS) 104
- CDRSC statements
 - API 149
 - NCF 144
- CDs
 - Dynamic Workload Console
 - installation 252
- checklist for installing 44
- checkpoint data set (EQQCKPT) 109
 - considerations when allocating 112
- class of service table 146
- CLIST library 109, 136
- Cloud & Smarter Infrastructure technical training xiv
- COFDLFnn member of SYS1.PARMLIB 91
- commands
 - TSO
 - BACKUP 90
 - BULKDISC 90
 - OPINFO 90
 - OPSTAT 90
 - SRSTAT 90
 - WSSTAT 90
- COMMNDnn member of SYS1.PARMLIB 91
- configurations
 - connecting IBM Workload Scheduler for z/OS systems 18
 - cross-system coupling facility (XCF) 19
 - description 9
 - event data set 17
 - examples
 - introduction 23, 371
 - NCF connection 381, 384
 - PLEX 379
 - server 21
 - shared DASD connection 23, 371, 374
 - single address space 29
 - sysplex 376
 - TCP/IP connection 26, 382
 - VTAM connection 24, 381, 384
 - XCF connection 27, 376
 - MAS (Multi-Access Spool)
 - restrictions 21
 - Multi-Access Spool (MAS)
 - restrictions 21
 - NCF (network communication function) 19
 - planning 17
 - shared DASD 19
 - VTAM 19
 - workload restart
 - description 20

- configurations (*continued*)
 - workload restart (*continued*)
 - examples 372, 375, 378, 381, 383, 385
 - workstation destination 20
 - XCF (cross-system coupling facility) 19
- connection
 - to IBM Workload Scheduler 274
- console
 - portfolio 279
 - start 279
- controlled systems
 - description 9
 - software requirements 4
- controller
 - checking the message log 168
 - description 6
 - IBMOPC resource class 95
 - loading national language support (NLS) software 56
 - loading software 55
 - RACF 93, 95
 - software requirements 4
 - started task data sets 130, 132
 - verifying installation 167
- controller, migrating with IWSZSELFUPGRADE 203
- controlling system
 - description 9
- COS table 146
- COUPLEnn member of SYS1.PARMLIB 89
- createNewWasProfile response file property 396
- creating sample JCL 58
- creating Workload Automation Programming Language samples 78
- critical job table data set (EQQJTABL) 109
- cross-domain resource definitions
 - API 149
 - NCF 144
- cross-system coupling facility (XCF)
 - groups 141
 - including 141
 - initialization statements 143
 - MVS initialization options 89
 - run time options 142
- current plan data set (EQQCPnDS) 103
 - considerations when allocating 107
- current plan data set (EQQSTDS and EQQNSTDS) 108

D

- data
 - considerations when allocating
 - dual job-tracking log (EQQDLnn) 116
 - extended-auditing archive (EQQDBARC) 117
 - extended-auditing log (EQQDBnn) 117
 - job-tracking archive (EQQJTARC) 116
 - job-tracking log (EQQJTnn) 116

- data lookaside facility (DLF) 91
- data set triggering
 - implementing 92
- data set triggering selection table macro (EQQLSENT)
 - invoking 391
 - syntax 391
- data sets
 - allocating 102
 - VSAM 102
 - considerations when allocating (event-driven workload)
 - automation configuration file (EQQEVLIB) 115
 - application description (EQQADDS) 106
 - archive of current plan (EQQACPDS) 106
 - automatic-recovery-procedure library (EQQPRLIB) 121
 - checkpoint (EQQCKPT) 112
 - current plan (EQQCPnDS) 107
 - diagnostic (EQQDUMP) 112
 - diagnostic message and trace (EQQDMSG) 112
 - dump (SYSDUMP) 113
 - event (EQQEVdnn) 113
 - event (EQQEVDS) 113
 - event (EQQHTTPO) 113
 - extended data (EQQXDnDS) 108
 - extended-auditing archive (EQQDBARC) 117
 - general 102
 - JCC incident log 116
 - JCC incident work (EQQINCWK) 116
 - JCC message table (EQQJCLIB) 115
 - JCL repository (EQQJSnDS) 107
 - job library (EQQJBLIB) 115
 - job-tracking archive (EQQJTARC) 116
 - loop analysis (EQQLOOP) 119
 - message log (EQQMLOG) 119
 - parameter library (EQQPARM) 120
 - PIF parameter data set (EQQYPARM) 120
 - started-task submit (EQQSTC) 121
 - submit/release (EQQSUDS) 121
 - in Tivoli OPC procedure JCL
 - optional 132
 - required 130
 - ISPF profile 136
 - migrating 194, 198, 199
 - security 94
 - step awareness
 - current plan (EQQSTDS and EQQNSTDS) 108
 - Tivoli OPC dialog 139
- databases, migrating 194, 198, 199
- datasets
 - allocating non-VSAM 109
- DB2
 - migrating history 191
 - migrating reporting 191

- DB2 (*continued*)
 - setup to run reports on DWC 128
 - DB2 database
 - creating 134
 - DB2 for DWC reports
 - setting up 126
 - default dialog-controller connection
 - table 137
 - diagnostic data set
 - EQQDMSG 109, 112
 - EQQDUMP 109, 112, 146
 - SYSDUMP 88, 110, 113
 - dialog
 - description 13
 - entering 140
 - EQQMINOx, authorizing 90
 - initializing 136
 - ISPF command table 136
 - problem determination
 - procedures 169
 - security 95
 - directories created outside of TWA_home
 - when installing Dynamic Workload Console 251
 - when installing z/OS connector 251
 - dispatching priority 91
 - distribution tape 14
 - DLF (data lookaside facility) 91
 - dockerfile 315
 - dump content definitions 88
 - dump data set (SYSDUMP) 88, 110, 113
 - DVDs
 - installation 213
 - dynamic domain manager installation
 - scanning system prerequisites 256
 - dynamic exits (PROGnn) 86, 87
 - dynamic workload broker
 - available functions 273
 - server connection 274
 - Dynamic Workload Console
 - accessibility xiv
 - activating 155
 - activating support for 155
 - configuration 277
 - connection
 - to dynamic workload broker
 - components 274
 - directories created at installation time
 - outside of TWA_home 251
 - getting started 279
 - installation
 - CDs 252
 - images 252
 - log files 270
 - methods 259
 - setup file 252
 - installation and uninstallation log
 - files 301
 - log files 270
 - overview 249
 - prerequisite 255
 - remove
 - manually 301, 302
 - server support 155
 - starting and stopping 275
 - troubleshooting 301
 - Dynamic Workload Console (*continued*)
 - uninstall 299
 - clean-up 301, 302
 - in silent mode 299
 - manually 301, 302
 - using wizard 299
 - upgrading 282, 285, 288, 296
 - multiple instances 292
 - overview 281
 - user interface 272
 - Dynamic Workload Console and z/OS
 - connector
 - secure communications 238
- E**
- ECSA (extended common service area) 84
 - education xiv
 - EDWA configuration file repository (EQQEVLIB) 109
 - end-to-end
 - FTA 10
 - end-to-end centralized script data set (EQQTWSCS) 122
 - End-to-end data set for centralized script support (EQQTWSCS) 110
 - end-to-end event data sets (EQQTWSIN/OUT) 110
 - end-to-end input events data set (EQQTWSIN) 122
 - end-to-end output events data set (EQQTWSOU) 122
 - end-to-end script library (EQQSCLIB) 110
 - ended-in-error-list layout table 138
 - environment setup 361
 - EQQACPDS (archive of current plan)
 - considerations when allocating 106
 - EQQADDS (application description data set) 102
 - considerations when allocating 106
 - EQQBKPT (backup checkpoint data set) 109
 - EQQBRDS (internal reader data set) 111
 - EQQCKPT (checkpoint data set) 109
 - considerations when allocating 112
 - EQQCPnDS (current plan data set) 103
 - considerations when allocating 107
 - EQQDBARC (extended-auditing-archive data set) 109
 - considerations when allocating 117
 - EQQDBnn (extended-auditing-log data set)
 - considerations when allocating 117
 - EQQDLnn (dual job-tracking-log data set) 109
 - considerations when allocating 116
 - EQQDMSG (diagnostic message and trace data set) 109
 - considerations when allocating 112
 - EQQDUMP (diagnostic data set) 109, 146
 - considerations when allocating 112
 - EQQEVDnn (event data set for an event reader) 109
 - calculating optimum LRECL 114
 - EQQEVDnn (event data set for an event reader) (*continued*)
 - considerations when allocating 113
 - EQQEVDs (event data set) 109
 - calculating optimum LRECL 114
 - considerations when allocating 113
 - EQQEVLIB (EDWA configuration file repository) 109
 - EQQEVLIB (event-driven workload automation configuration file data set)
 - considerations when allocating 115
 - EQQEXIT (event-tracking-code generation macro)
 - in JES exits 81, 387
 - in SMF exits 81, 387
 - invoking 387
 - syntax 388
 - EQQHTTP0 (event data set) 109
 - considerations when allocating 113
 - EQQICNVH (sample to migrate the history database) 135
 - EQQINCWK (JCC incident work data set) 109
 - considerations when allocating 116
 - EQQINITK 84
 - EQQJBLIB (job library data set) 109
 - considerations when allocating 115
 - EQQJCLIB (JCC message table data set) 109
 - considerations when allocating 115
 - EQQJOBS installation aid
 - creating sample JCL 58
 - creating Workload Automation Programming Language
 - samples 78
 - description 56
 - generating batch-job skeletons 69
 - setting up 57
 - EQQJSnDS (JCL repository data set) 103
 - considerations when allocating 107
 - EQQJTABL (critical job table data set) 109
 - EQQJTARC (job-tracking-archive data set) 109
 - considerations when allocating 116
 - EQQJTnn (extended-auditing-log data set) 109
 - EQQJTnn (job-tracking-log data set) 109
 - considerations when allocating 116
 - EQQLDDS (long-term plan work data set) 103
 - EQQLOOP (loop analysis data set) 109
 - EQQLOOP (loop analysis log data set)
 - considerations when allocating 119
 - EQQLSENT (data set triggering selection table macro)
 - invoking 391
 - syntax 391
 - EQQLTBKP (long-term-plan backup data set) 103
 - EQQLTDS (long-term plan data set) 103
 - EQQMINOx, authorizing for TSO 90
 - EQQMLOG 42
 - checking at the controller 168
 - checking at the server 168
 - EQQMLOG (message log data set) 109
 - considerations when allocating 119

EQQMLOG2 42
 EQQMONDS (monitoring data set) 109
 EQQNCPDS (new current plan data set) 103
 EQQNCXDS (new current plan extension data set) 103
 EQQNSTDS (current plan data set) 108
 EQQOCPBK (data set) 110
 EQQOIDS (operator instruction data set) 103
 EQQPARM (parameter library) 110
 considerations when allocating 120
 EQQPRLIB (automatic-recovery-procedure library) 110
 considerations when allocating 121
 EQQRDDS (resource description data set) 103
 EQQSCLIB (end-to-end script library) 110
 EQQSIDS (side information data set) 103
 EQQSTC (started-task-submit data set) 110
 considerations when allocating 121
 EQQSTDS (current plan data set) 108
 EQQSUDS (submit/release data set) 19, 110
 considerations when allocating 121
 EQQTROUT (input to EQQAUDIT) 110
 EQQTROUT (tracklog data set) 117
 EQQTWSCS (end-to-end centralized script dataset) 122
 EQQTWSCS (End-to-end data set for centralized script support) 110
 EQQTWSIN (end-to-end input events data set) 122
 EQQTWSIN/OUT (end-to-end event data sets) 110
 EQQTWSOU (end-to-end output events data set) 122
 EQQUPGBL sample 203
 EQQUPGWA sample 203
 EQQWSDS (workstation and calendar data set) 104
 EQQXDnDS (extended data, data set)
 considerations when allocating 108
 EQQYPARM (parameter data set)
 considerations when allocating 120
 EQQYPARM (PIF parameter data set) 110
 event data set
 description 17
 for an event reader (EQQEVDDnn) 109
 for an event writer (EQQEVDS) 109
 for submit checkpointing 17, 109
 verify 164
 event types 163
 event writer
 using with event reader function 42
 event-driven workload automation
 configuration file data set (EQQEVLIB)
 considerations when allocating 115
 event-tracking-code generation macro (EQQEXIT)
 in JES exits 81, 387
 in SMF exits 81, 387

event-tracking-code generation macro (EQQEXIT) (*continued*)
 invoking 387
 syntax 388
 exits
 event tracking 81
 extended common service area (ECSA) 84
 extended data (EQQXDnDS)
 considerations when allocating 108
 extended-auditing data sets
 considerations when allocating 117
 extended-auditing-archive data set (EQQDBARC) 109
 extended-auditing-log data set (EQQTnn) 109

F

fallback 207
 feature response file property 398
 files
 /etc/password 320
 FIPS support 158
 firewall stopping installation of the Dynamic Workload Console 305
 functions, IBM Workload Scheduler for z/OS
 subtasks, activating 11

G

GDDM 138
 generating batch-job skeletons 69
 graphical attribute table 138
 gskkyman 99, 231

H

hardware requirements 3
 Hiperbatch support
 COFDLFnn, updating 91
 history function
 installing 134
 hot standby 41

I

IBM i
 agent dynamic 335, 341
 IBM i IBM Workload Scheduler for z/OS
 Agent 310
 IBM Workload Scheduler 249
 available functions 273
 engine connection 274
 IBM Workload Scheduler for Applications 249
 IBM Workload Scheduler for z/OS 249
 IBM Workload Scheduler for z/OS Agent
 uninstalling 323
 upgrading 317
 IBM Workload Scheduler for z/OS Agent
 uninstalling
 twinsinst 323

IBM Workload Scheduler scanning
 system prerequisites for IBM Workload Scheduler 256
 IBMOPC resource class 95
 ICHRIN03 93
 id response file property 395, 396, 398
 IKJTSONn member of SYS1.PARMLIB
 AUTHCMD statement, updating for Tivoli OPC TSO commands 90
 AUTHTSF statement, updating for EQQMINOX 90
 images
 Dynamic Workload Console
 installation 252
 incident log (JCC)
 considerations when allocating 116
 initialization statements
 defining 134
 input to EQQAUDIT (EQQTROUT) 110
 install
 Java runtime 310, 317, 337, 342
 verifying software prerequisites 255
 installation 251
 bundles 213
 checking prerequisites IBM i 333
 Dynamic Workload Console
 CDs 252
 images 252
 in silent mode 259
 methods 259
 setup file 252
 using launchpad 259
 using wizard 259
 Dynamic Workload Console log files 301
 fails (Dynamic Workload Console) 305
 log files, Jazz for Service Management extension for WebSphere 272
 scanning system prerequisites for IBM Workload Scheduler 256
 scanning system prerequisites for Installation Manager 257
 silent 264, 265
 Dynamic Workload Console response file template 266
 of the Dynamic Workload Console, problems with 305
 troubleshooting scenarios
 Dynamic Workload Console 304
 twinsinst
 troubleshooting 329
 Installation
 DVD 252
 eImages 253
 HTTP or HTTPS Web server 253
 on your workstation 253
 repository 253
 installation agent
 return code 327, 349
 installation and uninstallation log files
 twinsinst 315, 320, 324
 z/OS connector 237
 Installation Manager
 installation and uninstallation log files 270
 packaging 271

- Installation Manager (*continued*)
 - wizard 271
- Installation Manager scanning
 - system prerequisites for Installation Manager 257
- installation method
 - twsinst 310
- installation methods
 - z/OS connector on WebSphere Application for z/OS 233
- installFixes response file property 398
- installing
 - authorization roles z/OS connector on WebSphere Application for z/OS 233
 - availability 41
 - checklist 44
 - considerations 41
 - DB2 database 134
 - detailed instructions 53
 - EQQJOBS installation aid
 - creating sample JCL 58
 - creating Workload Automation Programming Language samples 78
 - description 56
 - generating batch-job skeletons 69
 - setting up 57
 - event tracking exits 81
 - hot standby 41
 - initialization statements, defining 134
 - loading controller software 55
 - loading national language support (NLS) software 56
 - loading tracker software 55
 - MAS (Multi-Access Spool)
 - restrictions 21
 - Multi-Access Spool (MAS)
 - restrictions 21
 - NCF 143
 - overview 15
 - planning 41
 - RACF 93
 - security 93
 - started-task operations, implementing support for 129
 - verifying 161
 - configuration 174
 - controller 167
 - standby controller 170
 - submit events 183
 - tracker 161
 - tracking events 163
 - z/OS connector in silent mode 235
 - z/OS connector on WebSphere Application for z/OS 233
- installing additional plug-ins on z/OS connector on WebSphere Application Server for z/OS
 - installing additional plug-ins 238
- installing on additional plug-ins on WebSphere Application Server for z/OS 238
- installing on WebSphere Application for z/OS
 - z/OS connector using the Integrated Solutions Console 234

- installing on WebSphere Application for z/OS (*continued*)
 - z/OS connector using zConnInstall.sh 235
- installLocation response file
 - property 396
- Integrated Solutions Console
 - installing the z/OS connector 234
 - uninstalling the z/OS connector 242
- interactive installation
 - problem using with the Dynamic Workload Console 305
- interactive wizard
 - problem using with the Dynamic Workload Console 305
- internal reader data set (EQQBRDS) 111
- ISPF (Interactive System Productivity Facility)
 - command table 136
 - table library 136
- IWSZSELFUPGRADE, application to migrate controller 203

J

- Java runtime
 - installation 310, 317, 337, 342
- Java utilities
 - activating support for 157
- JCL repository data set (EQQJSnDS) 103
 - considerations when allocating 107
- JES exits, installing 81
- JES2
 - EXIT51 387
 - EXIT7 387
 - MAS (Multi-Access Spool)
 - restrictions 21
 - Multi-Access Spool (MAS)
 - restrictions 21
- JES3
 - IATUX19 387
 - IATUX29 387
- JESJOBS RACF resource class 97
- JESSPOOL RACF resource class 98
- job completion checker (JCC)
 - data sets
 - considerations when allocating 115
 - incident log 110, 116
 - incident work 116
 - incident work (EQQINCWK) 109
 - message table (EQQJCLIB) 109, 115
 - job library data set (EQQJBLIB) 109
 - considerations when allocating 115
- job log 9
- job-tracking data sets
 - considerations when allocating 116
 - dual job-tracking-log data set (EQQDLnn) 109
 - job-tracking-archive data set (EQQJTARC) 109
 - job-tracking-log data set (EQQJTnn) 109
 - tracklog data set (EQQTROUT) 117

- joblog and Restart Information requests
 - log data sets
 - EQQLOGRC (joblog and Restart Information requests log data sets) 109

L

- language packs
 - installing 313, 319, 339, 343
- launchpad
 - problems using with the Dynamic Workload Console 304
- libraries
 - CLIST 136
 - ISPF table 136
 - link 88
 - sample (SEQQSAMP) 355
 - link library (LNKLSTnn) 88
- Linux
 - erroneous warning messages
 - displayed from launchpad 304
 - LNKLSTnn member of
 - SYS1.PARMLIB 88
 - load module library (IEAAPFnn) 86
 - log file
 - z/OS connector 237
 - log files
 - Dynamic Workload Console 301
 - Jazz for Service Management extension for WebSphere installation 272
 - log management 9
 - logon mode table 145
 - long-term plan data set (EQQLTDS) 103
 - long-term plan work data set (EQQLDDS) 103
 - long-term-plan backup data set (EQQLTBKP) 103
 - loop analysis data set
 - EQQLOOP 109
 - loop analysis log data set
 - EQQLOOP
 - considerations when allocating 119

M

- macros
 - EQQEXIT (event-tracking-code generation macro) 81, 387
 - EQQLSENT (data set triggering selection table macro) 391
- maintaining
 - agent 327
 - z/OS connector 245
- manually
 - Dynamic Workload Console
 - uninstall 301, 302
 - zConnector
 - uninstall 301, 302
- MAS (Multi-Access Spool)
 - restrictions 21
- master domain manager installation
 - scanning system prerequisites 256

- master domain manager upgrade
 - scanning system prerequisites 256
- MAXECSA values 85
- message log data set
 - checking at the controller 168
 - checking at the server 168
 - EQQMLOG 109
 - considerations when allocating 119
- methods
 - for uninstalling 325
- migrating
 - backup controller 192
 - controller 203
 - data sets 194, 198, 199
 - databases 194, 198, 199
 - DB2 history 191
 - DB2 reporting 191
 - EQQICTOP conversion program 194
 - overview 189
- MLOG switch 42
- modify response file property 398
- Monitoring data set (EQQMONDS) 109
- Multi-Access Spool (MAS)
 - restrictions 21
- multiple systems
 - configuration examples 371
- MVS
 - relationship with IBM Workload Scheduler for z/OS 13
 - router service 95
 - SSI (subsystem interface) 13
 - subsystem interface (SSI) 13

N

- national language support (NLS) 56
- NCF (network communication function)
 - activating 143
 - application node definitions 144
 - cross-domain resource definitions 144
- network communication function (NCF)
 - activating 143
 - application node definitions 144
 - cross-domain resource definitions 144
- new current plan data set (EQQNCPDS) 103
- new current plan extension data set (EQQNCXDS) 103
- NLS (national language support) 56

O

- offering id response file property 398
- operating systems
 - supported 255
- operator instruction data set (EQQOIDS) 103
- OPERCMDS RACF resource class 97
- OPINFO command 90
- OPSTAT command 90
- output collector 9, 155

- overview
 - upgrading
 - Dynamic Workload Console 281

P

- packaging
 - log files 271
- parallel sysplex
 - configuration examples 371
- parallel test 190
- parameter library (EQQPARM) 110
 - considerations when allocating 120
- parameter twsinst update
 - addjruntime 342
 - backup_dir 319
 - domain 319
 - inst_dir 319
 - lang 319
 - nobackup_dir 319
 - reset_perm 320
 - skip_usercheck 320
 - tdwbhostname 343
 - tdwbport 343
 - uname 319, 344
 - update 344
 - wait 320, 344
 - work_dir 320, 344
- performance considerations
 - dispatching priority 91
 - event data set (EQQEVDs), calculating optimum LRECL 114
 - program properties table (PPT) 91
 - starting an event writer with an event reader 42
- PIF parameter data set (EQQYPARM) 110
 - considerations when allocating 120
- planning
 - installation 41
- PLEX configuration 379
- portfolio
 - console 279
- PPT (program properties table) 91
- prerequisite
 - Dynamic Workload Console 255
- prerequisites
 - IBM i 333
- problem determination procedures
 - authority 170
 - dialog 169
 - job tracking 163
 - security 170
 - tracker 163
- procedure
 - for uninstalling 325
- profile response file property 398
- PROGnn member of SYS1.PARMLIB 86, 87
- program directory 14
- program properties table (PPT) 91
- program temporary fix (PTF) 14
- PTF (program temporary fix) 14

R

- RACF 13
 - APPL resource class 95
 - batch jobs 93
 - IBMOPC resource class 95
 - ICHRIN03 93
 - modifying 93, 95
 - STARTED resource class 93
 - started task 94
 - user ID of Tivoli OPC submitted jobs 93
 - using functions of RACF 1.9
 - JESJOBS resource class 97
 - JESSPOOL resource class 98
 - OPERCMDs resource class 97
 - SURROGAT resource class 97
 - using functions of RACF 2.1
 - STARTED resource class 93
- RACF security
 - Dynamic Workload Console and z/OS connector 238
- ready-list layout table 138
- RECEIVE processing 361
- related software 5
- remove
 - Dynamic Workload Console
 - manually 301, 302
 - zConnector
 - manually 301, 302
- removing the product
 - IBM Workload Scheduler for z/OS
 - Agent 323
 - twsinst 323, 345
- repository location 398
- requirements 3
 - hardware 3
 - software 4
- resource description data set (EQQRDDS) 103
- response file missing, causing silent installation to fail 305
- response files
 - Dynamic Workload Console 264
 - silent installation 264, 265
 - template 266
- restore agent
 - return code 327, 349
- return code
 - twsinst 327, 349
- roles
 - authorization z/OS connector on WebSphere Application for z/OS 233

S

- SAF (system authorization facility) 13, 95
- sample JCL, creating 58
- sample library (SEQQSAMP) 14, 355
- sample to migrate the history data base (EQQICNVH) 135
- scan

- system prerequisites for IBM Workload Scheduler 256

- scan (*continued*)
 - system prerequisites for Installation Manager 257
 - scanning
 - system prerequisites for IBM Workload Scheduler 256
 - system prerequisites for Installation Manager 257
 - SCHEDnn member of SYS1.PARMLIB 91
 - secure communications
 - Dynamic Workload Console and z/OS connector 238
 - RACF security 238
 - security 13
 - APPL resource class 95
 - batch jobs 93
 - IBMOPC resource class 95
 - modifying 93, 95
 - router table 95
 - SAF (system authorization facility) 95
 - STARTED resource class 93
 - started task 94
 - system authorization facility (SAF) 95
 - user ID of the Tivoli OPC address space 93
 - user ID of Tivoli OPC submitted jobs 93
 - using functions of RACF 1.9
 - JESJOBS resource class 97
 - JESSPOOL resource class 98
 - OPERCMD5 resource class 97
 - SURROGAT resource class 97
 - security, problem determination procedures 170
 - SEQQSAMP (sample library) 14, 355
 - server
 - activating 150
 - checking the message log 168
 - description 7
 - optional data sets 132
 - required data sets 130
 - sample configuration 379
 - updating APPC/MVS options 153
 - setup file
 - Dynamic Workload Console installation 252
 - side information data set (EQQSIDS) 103
 - silent installation 264, 265
 - of the Dynamic Workload Console 259
 - response file template 266
 - silent installation of the Dynamic Workload Console problems with 305
 - silent mode
 - installing the z/OS connector 235
 - silent uninstall
 - of the Dynamic Workload Console 299
 - silent upgrade
 - response file template 266
 - SMF exits, installing 81
 - SMF parameters (SMFPRMnn) 86, 87
 - SMFPRMnn member of SYS1.PARMLIB 86, 87
 - SRSTAT command 90
 - SSL
 - controller 99
 - server 99
 - TCPOPTS 99
 - z/OS connector and server started task 231
 - standby controller
 - verifying installation 170
 - Standby controller
 - description 10
 - STARTED resource class 93
 - started-task operations, implementing support for 129
 - started-task procedure
 - controller 128
 - output collector 128
 - tracker 128
 - started-task-submit data set (EQQSTC) 110
 - considerations when allocating 121
 - starting
 - console 279
 - Dynamic Workload Console server 275
 - step awareness (EQQSTDS and EQQNSTDS) 108
 - stopping
 - Dynamic Workload Console server 275
 - structure
 - installation DVD structures 213
 - submit checkpointing 17
 - submit/release data set (EQQSUDS) 19, 110
 - considerations when allocating 121
 - subsystem
 - APPL resource class 95
 - name table (IEFSSNnn) 83
 - subtasks, activating 11
 - supported
 - operating systems 255
 - SURROGAT RACF resource class 97
 - SYS1.PARMLIB
 - APPC/MVS options (APPCPMnn) 91, 149, 153
 - defining subsystems (IEFSSNnn) 83
 - dispatching priority 91
 - dynamic exits (PROGnn) 86, 87
 - EQQMINOx, authorizing for TSO (IKJTSONn) 90
 - Hiperbatch support (COFDLFnn) 91
 - link library (LNKLSTnn) 88
 - load module library (IEAAPFnn) 86, 88
 - performance (SCHEDnn) 91
 - program properties table (PPT) 91
 - SMF parameters (SMFPRMnn) 86
 - starting Tivoli OPC (COMMNDnn) 91
 - SYSMDUMP 88
 - updating dump content definitions 88
 - XCF initialization options (COUPLEnn) 89
 - SYS1.PROCLIB
 - controller 128
 - tracker 128
 - SYSMDUMP (dump data set) 88, 110
 - considerations when allocating 113
 - sysplex
 - configuration examples 371
 - sysplex (system complex) 376
 - system authorization facility (SAF) 13, 95
 - system complex (sysplex) 376
 - system prerequisites
 - scan for IBM Workload Scheduler 256
 - system prerequisites for Installation Manager
 - scan 257
- ## T
- TCPOPTS 231
 - technical training xiv
 - Tivoli Dynamic Workload Console
 - configuration 277
 - overview 249
 - starting and stopping 275
 - troubleshooting 301
 - upgrading
 - overview 281
 - Tivoli OPC server
 - activating support for 150
 - server support 150
 - Tivoli Workload Scheduler agents IBM i
 - uninstalling
 - twinsinst 345
 - Tivoli Workload Scheduler for z/OS
 - Agent 310
 - tracker
 - description 6
 - loading software 55
 - optional data sets 132
 - RACF 93
 - started task data sets 130
 - verifying installation 161
 - tracklog data set (EQQTROUT) 117
 - training
 - technical xiv
 - troubleshooting
 - agent 327
 - installation scenarios
 - Dynamic Workload Console 304
 - z/OS connector 245
 - TSO
 - IKJTSONn member of SYS1.PARMLIB
 - AUTHCMD statement, updating for Tivoli OPC TSO commands 90
 - AUTHTSF statement, updating for EQQMINOx 90
 - RACF user 95
 - TSO commands
 - BACKUP 90
 - BULKDISC 90
 - OPINFO 90
 - OPSTAT 90
 - SRSTAT 90
 - WSSTAT 90

- twinst 310, 335, 341
 - authorization roles 310
 - installation and uninstallation log files 315, 320, 324, 347
 - installation method 310
 - return code 327, 349
 - uninstalling 323, 345
 - usage 310

U

- uninstall
 - Dynamic Workload Console 299
 - manually 301, 302
 - of the Dynamic Workload Console in silent mode 299
 - using response file 299
 - zConnector
 - manually 301, 302
- uninstallation
 - Dynamic Workload Console log files 301
- uninstallation agent
 - return code 327, 349
- uninstallation procedures 325
- uninstalling
 - additional plug-ins 325
 - authorization roles z/OS connector on WebSphere Application for z/OS 233
 - IBM Workload Scheduler for z/OS Agent 323
 - procedures 325
 - z/OS connector on WebSphere Application for z/OS 233
- uninstalling additional plug-ins on z/OS connector on WebSphere Application Server for z/OS
 - uninstalling additional plug-ins 243
- uninstalling agent
 - twinst 345
- uninstalling IBM Workload Scheduler for z/OS Agent
 - twinst 323
- uninstalling on additional plug-ins on WebSphere Application Server for z/OS 243
- uninstalling on WebSphere Application Server for z/OS
 - z/OS connector using the Integrated Solutions Console 242
 - z/OS connector using zConnUninstall.sh 242
- updateWas, using to update the SOAP properties after changing application server user or password 245
- upgrade
 - scanning system prerequisites 256
 - silent
 - Dynamic Workload Console response file template 266
- upgrade agent
 - return code 327, 349
- upgrading
 - Dynamic Workload Console 282, 285, 288, 296
 - overview 281

- upgrading (*continued*)
 - IBM Workload Scheduler for z/OS Agent 317
- usage
 - twinst 310
- user.offeringId response file
 - property 396, 399
- user.wasInstallLocation response file
 - property 396
- user.wasPassword response file
 - property 397, 399
- user.wasProfileLocation response file
 - property 396
- user.wasUserName response file
 - property 396, 399
- user.zosConnEngineName 397
- user.zosConnHostname 397
- user.zosConnIsEnabled 397
- user.zosConnPort 397
- user.zosConnSslIsEnabled 397

V

- VARY ACT command 146
- verifying installation 161
 - configuration 174
 - controller 167
 - standby controller 170
 - submit events 183
 - tracker 161
 - tracking events 163
- verifying software prerequisites
 - install 255
- VTAM
 - activate network resources 146
 - class of service table 146
 - defining NCF 144
 - logon mode table 145
 - VARY ACT command 146

W

- WebSphere Application for z/OS
 - installing the z/OS connector using zConnInstall.sh 235
- WebSphere Application for z/OS
 - installing
 - z/OS connector 233
- WebSphere Application for z/OS
 - uninstalling
 - z/OS connector on 233
- WebSphere Application for z/OS
 - upgrading
 - z/OS connector on 233
- WebSphere Application Server for z/OS
 - installing additional plug-ins on z/OS connector 238
 - uninstalling additional plug-ins on z/OS connector 243
 - uninstalling the z/OS connector using zConnUninstall.sh 242
- Windows
 - undefined error message displayed from launchpad 304
- wizard
 - log files 271

- Workload Automation Programming Language
 - creating samples 78
 - environment setup 128
- workload restart (WLR)
 - description 20
 - examples 372, 375, 378, 381, 383, 385
- workstation
 - destination 20
- workstation and calendar data set (EQQWSDS) 104
- WSSTAT command 90

X

- XCF (cross-system coupling facility)
 - groups 141
 - including 141
 - initialization statements 143
 - MVS initialization options 89
 - run time options 142

Z

- z-centric
 - log management 9
- z/OS connector
 - applying maintenance 241
 - authorization roles installing on WebSphere Application for z/OS 233
 - authorization roles uninstalling on WebSphere Application for z/OS 233
 - authorization roles upgrading on WebSphere Application for z/OS 233
 - directories created at installation time
 - outside of TWA_home 251
 - installation and uninstallation log files 237
 - installing on WebSphere Application for z/OS 233
 - maintaining 245
 - on WebSphere Application Server for z/OS business scenario 233
 - troubleshooting 245
 - uninstalling on WebSphere Application for z/OS 233
 - upgrading
 - multiple instances 214
 - upgrading on WebSphere Application for z/OS 233
- z/OS connector and Dynamic Workload Console
 - secure communications 238
- z/OS connector on WebSphere Application for z/OS
 - installation methods 233
 - installing with the Integrated Solutions Console 234
 - installing with zConnInstall.sh 235
- z/OS connector on WebSphere Application Server for z/OS
 - secure communications 238

- z/OS connector on WebSphere Application Server for z/OS *(continued)*
 - uninstalling with the Integrated Solutions Console 242
 - uninstalling with zConnUninstall.sh 242
- z/OS connector on zWAS
 - secure communications 238
- zConnector
 - remove
 - manually 301, 302
 - uninstall
 - clean-up 301, 302
 - manually 301, 302
- zConnInstall.sh
 - installing the z/OS connector 235
- zConnUninstall.sh
 - uninstalling the z/OS connector 242



Product Number: 5698-T08

Printed in USA