

z/OS



MVS Planning: Workload Management

Version 2 Release 2

Note

Before using this information and the product it supports, read the information in "Notices" on page 263.

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

© **Copyright IBM Corporation 1994, 2015.**

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

Contents

Figures	vii
--------------------------	------------

Tables	ix
-------------------------	-----------

About this information	xi
---	-----------

Who should use this information	xi
Where to find more information	xi
Other referenced documents.	xii
Information updates on the web	xii

How to send your comments to IBM	xiii
---	-------------

If you have a technical problem	xiii
---	------

Summary of changes	xv
-------------------------------------	-----------

Changes made in z/OS Version 2 Release 2 (V2R2) as updated March 2017	xv
Changes made in z/OS Version 2 Release 2 (V2R2) as updated December 2015	xv
Summary of changes for z/OS Version 2 Release 2 (V2R2).	xvi
Changes made in z/OS Version 2 Release 1 (V2R1) as updated December 2013	xvi
Changes made in z/OS Version 2 Release 1	xvi

Chapter 1. Why MVS workload management?	1
--	----------

Problems addressed by workload management	1
MVS workload management solution for today and tomorrow	2

Chapter 2. What is MVS workload management?	3
--	----------

Performance administration	4
Performance management	4
Workload balancing	5
Workload management concepts.	5
What is a service definition?	5
Why use service policies?	6
Organizing work into workloads and service classes	6
Why use resource groups?	8
Assigning work to a service class	10
Why use application environments?	11
Why use scheduling environments?	12
Summary of service definition and service policy concepts	14

Chapter 3. Workload management participants.	17
---	-----------

Workload management work environments.	17
Subsystem support for goal types and multiple periods	18
Subsystem-specific performance hints.	19

Workload balancing	19
Workload management in a CPSM environment	20
Workload management in a DB2 Distributed Data Facility environment	21
Batch workload management	21
Multisystem enclave support	22
Intelligent Resource Director.	22
HiperDispatch mode	23
Overview	23
The concept of HiperDispatch mode	23
Setting HiperDispatch mode in SYS1.PARMLIB	24
I/O storage management.	25
Handling service class periods with a response time goal	25
Handling service class periods with a velocity goal	26
Handling other I/O requests	26
Controlling the information passed to the I/O manager	26
Non-z/OS partition CPU management	27
Workload management and Workload License Charges	27
Defining the capacity of a partition	28
Defining group capacity	29
Workload management with other products	32

Chapter 4. Setting up a service definition.	33
--	-----------

Specifying a service definition	33
Storing service definitions	34
Defining the parts of a service definition	34

Chapter 5. Defining service policies	35
---	-----------

Using policy overrides.	36
---------------------------------	----

Chapter 6. Defining workloads	39
--	-----------

Defining a departmental workload	39
--	----

Chapter 7. Defining resource groups	41
--	-----------

Calculating an LPAR share — Example 1	45
Specifying the capacity as a number of CPs — Example 2.	47

Chapter 8. Defining service classes and performance goals	49
--	-----------

Velocity formula.	52
Defining performance goals	53
Determining system response time goals	53
Examples of service classes with response time goals	55
Defining velocity goals	56
Adjusting velocity goals based on samples included in velocity calculation.	56
Using velocity goals for started tasks.	57

Using discretionary goals	57
Using performance periods	58
Defining goals appropriate for performance periods	58
Using importance levels in performance periods	59
Chapter 9. Defining classification rules	61
Defining classification rules for each subsystem	63
Defining work qualifiers	67
Defining z/OS WLM classification rules for performance policies from the Unified Resource Manager	79
Defining special reporting options for workload reporting	82
Defining the order of classification rules	85
Defining a subsystem service class default	85
Organizing work for classification	89
Using masking notation	90
Using wildcard notation	90
Using the start position	91
Using groups	93
Using the system-supplied service classes	95
Chapter 10. Defining report classes	99
Chapter 11. Defining service coefficients and options.	101
Calculating the amount of service consumed	101
Service definition coefficients	102
Changing your coefficient values	102
Using the storage (MSO) coefficient for calculations	103
Specifying I/O priority management	104
Considerations for I/O priority management	104
Enabling I/O priority groups	104
Considerations for I/O priority groups	104
Specifying dynamic alias management	105
Workload management considerations for dynamic alias management	105
HCD considerations for dynamic alias management	106
Chapter 12. Defining special protection options for critical work	109
Long-term storage protection	109
Storage critical for address spaces	109
Storage critical for CICS and IMS transactions	110
Long-term CPU protection	110
Long-term I/O protection	111
Honor priority	111
Modifications of transaction response time management	113
Sample scenarios	114
Scenario 1	114
Scenario 2	115
Scenario 3	117
Scenario 4	118
Scenario 5	119
Scenario 6	120
Reporting	121

Option summary	122
--------------------------	-----

Chapter 13. Defining application environments 125

Getting started with application environments	125
Specifying application environments to workload management	126
Selecting server limits for application environments	128
How WLM manages servers for an application environment	130
Using application environments	131
Managing application environments	132
Using operator commands for application environments	133
Making changes to the application environment servers	134
Changing the definition of an application environment	134
Handling error conditions in application environments	134
Authorizing application environment servers	135
Example for restricting access to application environment servers	135

Chapter 14. Defining scheduling environments 137

Getting started with scheduling environments	137
Specifying scheduling environments to workload management	138
Managing resource states	139
Associating scheduling environments with incoming work	143
Displaying information about scheduling environments and resource states	143
MVS operator commands	143
JES2/JES3 operator commands	145
SDSF commands	145

Chapter 15. Managing performance with the guest platform management provider (GPMP) 147

Overview of the guest platform management provider (GPMP)	147
Configuring the guest platform management provider on z/OS	148
Before You Begin	148
About this task	148
Procedure	149
Results	151
What to do next	151

Chapter 16. Workload management migration 153

Creating a service definition for the first time	153
Migrating to a new z/OS release with an existing service definition	155
Migration activities	156
Restricting access to the WLM service definition	156

Start the application and enter/edit the service definition	157
Calculate the size of the WLM couple data set	160
Allocate a WLM couple data set	160
Make a WLM couple data set available to the sysplex for the first time.	163
Make a newly formatted couple data set available to the sysplex	164
Migration considerations for velocity	165
Migration considerations for discretionary goal management	166
Migration considerations for dynamic alias management	166
Migration considerations for multisystem enclaves	166
Migration considerations for protection of critical work.	166
Migration considerations for managing non-enclave work in enclave servers.	167
Migration considerations for an increased notepad size.	167
WLM managed batch initiator balancing	168
Consider resource group maximum in WLM batch initiator management.	168

Chapter 17. Defining a coupling facility structure for multisystem enclave support 169

Defining the coupling facility	169
Shutting down the coupling facility	171
Coupling facility failures	171

Chapter 18. The Intelligent Resource Director 173

LPAR CPU management	175
Dynamic channel path management	175
Channel subsystem priority queuing	176
Example: How the Intelligent Resource Director works	176
Making the Intelligent Resource Director work	178
Defining the SYSZWLM_xxxxxxxx coupling facility structure	178
Enabling LPAR CPU management	179
Enabling non-z/OS CPU management	180
Enabling dynamic channel path management	181
Enabling channel subsystem priority queuing	182
For more information.	182

Chapter 19. Using System z Application Assist Processor (zAAP) . 185

Performing capacity planning to project how many zAAPs will be needed (zAAP Projection Tool)	185
Meeting software and hardware requirements associated with the zAAPs	186
Acquiring the zAAPs.	186
Defining zAAPs to the desired LPARs	187
Reviewing parameter settings associated with zAAP usage	187
Review z/OS parameter settings	187

Review Java parameter settings	188
Considering automation changes related to zAAP usage	188
Monitoring zAAP utilization and configuring changes appropriately	188

Chapter 20. Using System z Integrated Information Processor (zIIP) 191

Meeting software and hardware requirements for using zIIPs	191
Planning for zIIPs	191
Acquiring zIIPs.	192
Defining zIIPs	192
Reviewing z/OS parameter settings	193
Using zIIPs — miscellaneous services	193
Activating zIIPs	193

Chapter 21. Using the WLM ISPF application 195

Before you begin	195
Panel areas and how to use them.	196
Using the menu bar	196
Using the status line	197
Using the scrollable area.	197
Using the Action field	198
Using the command line	198
Using the function keys	199
Starting the WLM application	199
Now you're started	200
Using the Definition Menu	200
Using the menu bar on the Definition Menu	203
Working with service policies	205
Working with workloads	206
Working with resource groups.	207
Working with service classes	208
Defining goals	208
Using action codes on service class panels.	209
Defining service policy overrides	210
Working with classification rules	212
Using action codes on the Modify Rules panel	215
Using selection lists for classification rules.	218
Creating a subsystem type for rules	218
Deleting a subsystem type for rules	218
Working with classification groups	218
Working with report classes	219
Working with service coefficients and options	220
Working with application environments	221
Working with scheduling environments	222
Creating a new scheduling environment	222
Modifying a scheduling environment	224
Copying a scheduling environment	225
Browsing a scheduling environment.	226
Printing a scheduling environment	226
Deleting a scheduling environment	226
Creating a new resource.	227
Showing all cross-references for a resource definition.	230
Deleting a resource	231
Working with the guest platform management provider (GPMP) settings	232

Coordinating updates to a service definition	233
Using the WLM couple data set	233
Using MVS data sets	235
Restricting access to your service definition	235
Activating a service policy	236
Printing in the application	236
Browsing definitions	237
Using XREF function to view service definition relationships.	237
WLM application messages.	238

Chapter 22. Using the z/OS Management Facility (z/OSMF) to administer WLM 243

Overview of the z/OSMF workload management task	243
Key functions of the Workload Management task in z/OSMF	243

Appendix A. Customizing the WLM ISPF application 247

Specifying the exits	247
Coding the WLM exits	248
IWMARIN1	248
Customizing the WLM application libraries — IWMAREX1	248
Customizing the WLM application data sets — IWMAREX2	249

Adding WLM as an ISPF menu option	252
Moving pop-up windows	253
Customizing the keylists	254

Appendix B. CPU capacity table 255

Using SMF task time	255
-------------------------------	-----

Appendix C. Return codes for the IWMINSTL sample job. 257

Appendix D. Accessibility 259

Accessibility features	259
Consult assistive technologies	259
Keyboard navigation of the user interface	259
Dotted decimal syntax diagrams	259

Notices 263

Policy for unsupported hardware.	264
Minimum supported hardware	265
Trademarks	265

Workload management terms 267

Index 273

Figures

1.	MVS Workload Management Overview	3
2.	Workload organized by subsystem	8
3.	Workload organized by department	8
4.	Resource groups	9
5.	Work classification	11
6.	Application environment example	12
7.	Scheduling environment example	13
8.	Service definition, including two service policies	15
9.	Sysplex view of the management environment	18
10.	Workload management in a CICS environment	20
11.	Workload management enforcing a defined capacity limit	28
12.	Example of workload consumption for partitions MVS1, MVS2, and MVS3	31
13.	Example: Resource group overview	46
14.	Working With A Resource Type 2 - Sample Calculation.	47
15.	Using classification rules to assign work to service classes.	62
16.	Sample z/OS WLM Classification Rules Panel for Mapping Unified Resource Manager performance policies	81
17.	Formula for Calculating Service Consumption	101
18.	MSO Coefficient Formula	103
19.	Specifying the Storage Critical Option	109
20.	Specifying the CPU Critical option	110
21.	Specifying the I/O Priority Group option	111
22.	Specifying the High Priority option	112
23.	Specifying the Manage Region Using Goals Of option	113
24.	Scenario 1: Address Spaces	114
25.	Scenarios 2, 3, 4, 5: CICS/IMS regions	115
26.	Scenario 6: CICS Regions Adhering to a Work Manager/Consumer Model	120
27.	Sample Systems and Scheduling Environments	142
28.	One LPAR cluster on one CPC	173
29.	Four LPAR clusters on two CPCs	174
30.	Intelligent Resource Director example – Day shift.	177
31.	Intelligent Resource Director Example – Night Shift	178
32.	Menu Bar on the Definition Menu	196
33.	Definition Menu File Choices	196
34.	Service Class Selection List	197
35.	Service Class Selection List panel	198
36.	Action field on the Subsystem Type Selection List panel.	198
37.	Function key area	199
38.	Choose Service Definition pop-Up	200
39.	Definition Menu panel	201
40.	Create a Service Policy panel	205
41.	Service Policy Selection List panel.	206
42.	Create a Workload panel.	206
43.	Workload Selection List panel	207
44.	Create a Resource Group panel	207
45.	Create a Service Class panel	208
46.	Choose a Goal Type pop-up	209
47.	Average Response Time Goal pop-up	209
48.	Create a Service Class panel	209
49.	Action Codes for Goal	210
50.	Service Policy Selection List panel.	210
51.	Override Service Class Selection List panel	211
52.	Override Attributes for a Service Class panel	212
53.	Subsystem Type Selection List for Rules panel	213
54.	Modify Rules for the Subsystem Type panel	213
55.	Modify Rules for the Subsystem Type panel, scrolled right to description fields.	214
56.	Modify Rules for the Subsystem Type panel, scrolled right to Storage Critical, Reporting Attribute, and Manage Regions Using Goals Of fields	215
57.	Action codes for classification rules	215
58.	Create a Group panel	219
59.	Modify Rules for STC Subsystem	219
60.	Creating a report class on the Modify Rules panel	220
61.	Create a Report Class confirmation panel	220
62.	Service Coefficients panel	221
63.	Create an Application Environments panel	222
64.	Decide What to Create panel	222
65.	Scheduling Environment Selection List panel	223
66.	Create a Scheduling Environment panel	223
67.	Resource Definition List panel	223
68.	Create a Scheduling Environment panel	224
69.	Scheduling Environment Selection List panel	225
70.	Modify a Scheduling Environment panel	225
71.	Copy a Scheduling Environment panel	226
72.	Browse a Scheduling Environment panel	226
73.	Delete a Scheduling Environment panel	227
74.	Scheduling Environment Selection List panel	227
75.	Resource Definition List panel	228
76.	Define Resource oanel	228
77.	Resource Definition List panel	228
78.	Modify a Scheduling Environment panel	229
79.	Resource Definition List panel	229
80.	Resource Definition List panel	229
81.	Modify a Scheduling Environment Panel	230
82.	Resource Definition List panel	231
83.	Resource Cross-Reference Of Scheduling Environments panel	231
84.	Resource Definition List panel	232
85.	Guest Platform Management Provider Settings - sample panel	232
86.	Overwrite Service Definition panel	233
87.	Allocate couple data set using CDS values panel	234
88.	Allocate couple data set panel using service definition values	234
89.	Policy Selection List panel to activate a service policy	236

90. Browse function from the Service Class Selection List	237	93. Example of adding WLM as an option on the ISPF menu	253
91. Service Class Subsystem Xref panel	237	94. Keylist Utility panel	254
92. z/OS Management Facility - Overview Panel	245		

Tables

1. Levels of processing	26	11. Application environment server characteristics	129
2. Example of definitions for MVS1, MVS2, and MVS3 partitions	29	12. z/OS execution environment variables for the guest platform management provider	149
3. Example: LPAR configuration	45	13. WLM libraries	157
4. Example: ELPMAX in sysplex WLMPLEX	45	14. Functionality levels for service definition	158
5. IBM-defined subsystem types	64	15. The current WLM couple data set format level	160
6. Enclave transactions, address space-oriented transactions, and CICS/IMS transactions.	66	16. Values to use in storage estimation formulas	170
7. Work qualifiers supported by each IBM-defined subsystem type	67	17. Menu bar options on the Definition Menu	203
8. Effects of WLMPAV settings on base and alias devices	106	18. Return codes from IWMMARIN0	247
9. Summary of options for storage protection, CPU protection, and exemption from transaction response time management	122	19. WLM Libraries	248
10. IBM-supplied Subsystems Using Application Environments	126	20. Keylist names and usage descriptions	254
		21. Return codes from IWMINSTL.	257

About this information

This information supports z/OS (5650–ZOS). This document contains information to help you convert to z/OS[®] workload management, to use workload management, and to make the most out of workload management.

Note: The z/OS workload management component is sometimes also called “Workload Manager.”

Who should use this information

This document is intended for the system programmers, system analysts, and systems engineers who are responsible for developing a conversion plan for z/OS workload management, and who are responsible for implementing z/OS workload management.

Where to find more information

Where necessary, this document references information in other documents, using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS V2R2 Information Roadmap*.

Title	Order number
<i>z/OS Common Information Model User's Guide</i>	SC34-2671
<i>z/OS Migration</i>	GA32-0889
<i>z/OS MVS Capacity Provisioning User's Guide</i>	SC34-2661
<i>z/OS MVS Initialization and Tuning Guide</i>	SA23-1379
<i>z/OS MVS Initialization and Tuning Reference</i>	SA23-1380
<i>z/OS MVS JCL User's Guide</i>	SA23-1386
<i>z/OS MVS JCL Reference</i>	SA23-1385
<i>z/OS MVS System Messages, Vol 5 (EDG-GFS)</i>	SA38-0672
<i>z/OS MVS System Messages, Vol 9 (IGF-IWM)</i>	SA38-0676
<i>z/OS MVS Programming: Workload Management Services</i>	SC34-2663
<i>z/OS MVS Setting Up a Sysplex</i>	SA23-1399
<i>z/OS MVS System Commands</i>	SA38-0666
<i>z/OS MVS System Management Facilities (SMF)</i>	SA38-0667
<i>z/OS Planning for Installation</i>	GA32-0890
<i>IBM z Systems Ensemble Workload Resource Group Management Guide</i>	GC27-2629
<i>IBM z Systems Ensemble Planning Guide</i>	GC27-2631
<i>IBM z/OS Management Facility Configuration Guide</i>	SC27-8419

Other referenced documents

Title	Order number
<i>CICS/ESA CICS-RACF Security Guide</i>	SC33-1185
<i>CICS/ESA Customization Guide</i>	SC33-1165
<i>CICS/ESA Dynamic Transaction Routing in a CICSplex</i>	SC33-1012
<i>CICS/ESA Performance Guide</i>	SC33-1183
<i>CICS/ESA Resource Definition Guide</i>	SC33-1166
<i>CICSplex SM Managing Workloads</i>	SC33-1807
<i>DB2 Administration Guide</i>	Varies by version
<i>DB2 Data Sharing: Planning and Administration</i>	Varies by version
<i>DB2 SQL Reference</i>	Varies by version
<i>DCF SCRIPT/VS User's Guide</i>	S544-3191
<i>IMS/ESA Administration Guide: System</i>	SC26-8730
<i>IMS/ESA Installation Volume 1 and Volume 2</i>	SC26-8736 SC26-8737
<i>Internet Connection Server User's Guide</i>	SC31-8204
<i>z/OS V2R2 ISPF Dialog Developer's Guide and Reference</i>	SC19-3619
<i>z/OS JES2 Initialization and Tuning Guide</i>	SA32-0991
<i>z/OS JES2 Initialization and Tuning Reference</i>	SA32-0992
<i>z/OS JES2 Installation Exits</i>	SA32-0995
<i>z/OS JES3 Initialization and Tuning Guide</i>	SA32-1003
<i>z/OS JES3 Initialization and Tuning Reference</i>	SA32-1005
<i>z/OS MVS Planning: APPC/MVS Management</i>	SA23-1388
<i>z/OS RMF Report Analysis</i>	SC34-2665
<i>z/OS RMF User's Guide</i>	SC34-2664
<i>z/OS SOMobjects Configuration and Administration Guide</i>	GC28-1851
<i>z/OS TSO/E Customization</i>	SA32-0976
<i>z/OS UNIX System Services Programming Tools</i>	SA23-2282
<i>z/OS UNIX System Services Planning</i>	GA32-0884
<i>WebSphere MQ Workflow Administration Guide</i>	SH12-6289
<i>PR/SM Planning Guide</i>	Varies by CPC
<i>SAP on DB2 UDB for OS/390 and z/OS: Planning Guide, 2nd Edition, SAP Web Application Server 6.20</i>	n/a
<i>Support Element Operations Guide</i>	SC28-6802

Information updates on the web

For the latest information, visit the WLM/SRM web page at:

<http://www.ibm.com/systems/z/os/zos/features/wlm/>

How to send your comments to IBM

We appreciate your input on this documentation. Please provide us with any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

Use one of the following methods to send your comments:

Important: If your comment regards a technical problem, see instead “If you have a technical problem.”

- Send an email to mhvrcfs@us.ibm.com.
- Send an email from the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>).

Include the following information:

- Your name and address
- Your email address
- Your phone or fax number
- The publication title and order number:
 - z/OS V2R2 MVS Planning: Workload Management
 - SC34-2662-05
- The topic and page number or URL of the specific information to which your comment relates
- The text of your comment.

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

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

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one or more of the following actions:

- Visit the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

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

Changes made in z/OS Version 2 Release 2 (V2R2) as updated March 2017

Changed

Several topics are updated for new function that allows WLM administrators to prevent the overflow of specialty-engine-intensive work to standard processors. This includes the new Honor Priority attribute for service classes, and the new Memory Limit attribute for resource groups. See these topics for details:

- Chapter 8, “Defining service classes and performance goals,” on page 49
- “Storage critical for address spaces” on page 109
- “Long-term CPU protection” on page 110
- “Long-term I/O protection” on page 111
- “Honor priority” on page 111
- “Review z/OS parameter settings” on page 187
- “Reviewing z/OS parameter settings” on page 193
- Chapter 7, “Defining resource groups,” on page 41
- “Working with resource groups” on page 207
- “Working with service classes” on page 208
- “Defining goals” on page 208
- “Defining service policy overrides” on page 210
- “Migrating to a new z/OS release with an existing service definition” on page 155.

Changes made in z/OS Version 2 Release 2 (V2R2) as updated December 2015

Changed

Several topics are updated for WLM support of mobile pricing. This support allows WLM administrators to classify transactions in the WLM service definition so that they can benefit from mobile application pricing. The updated topics include the following:

- “Defining special reporting options for workload reporting” on page 82
- “Defining classification rules for each subsystem” on page 63
- “Defining work qualifiers” on page 67
- “Client transaction name” on page 71
- “Connection type” on page 71
- “Transaction class / job class” on page 76
- “Long-term storage protection” on page 109
- “Modifications of transaction response time management” on page 113

- “Working with classification rules” on page 212.

Summary of changes for z/OS Version 2 Release 2 (V2R2)

The following changes are made for z/OS Version 2 Release 2 (V2R2). All technical changes for z/OS V2R2 are indicated by a vertical line to the left of the change.

New

- Return code 320 has been added to the IWMINSTL sample job. See Appendix C, “Return codes for the IWMINSTL sample job,” on page 257.

Changed

- Information about defining velocity goals has been updated. See “Defining velocity goals” on page 56.
- An update has been made to the WLM time interval. See “Workload management and Workload License Charges” on page 27.

Changes made in z/OS Version 2 Release 1 (V2R1) as updated December 2013

Changed

- The order number for z/OS Version 2 Release 1 (5650-ZOS) has been corrected.
- Message IWMAM099 has been added in “WLM application messages” on page 238.

Changes made in z/OS Version 2 Release 1

See the following publication for all enhancements to z/OS Version 2 Release 1:

- *z/OS Summary of Message and Interface Changes*, SA23-2300
- *z/OS Introduction and Release Guide*, GA32-0887
- *z/OS Planning for Installation*, GA32-0890
- *z/OS Migration*, GA32-0889

This document contains information previously presented in *z/OS MVS™ Planning: Workload Management*, SA22-7602-20 which supports z/OS Version 1 Release 13.

Updated information

- Chapter 9, “Defining classification rules,” on page 61 and Chapter 16, “Workload management migration,” on page 153 were updated.

Chapter 1. Why MVS workload management?

Before the introduction of MVS workload management, MVS required you to translate your data processing goals from high-level objectives about what work needs to be done into the extremely technical terms that the system can understand. This translation requires high skill-level staff, and can be protracted, error-prone, and eventually in conflict with the original business goals. Multi-system, sysplex, parallel processing, and data sharing environments add to the complexity.

MVS workload management provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. MVS workload management is the combined cooperation of various subsystems (CICS®, IMS/ESA®, JES, APPC, TSO/E, z/OS UNIX System Services, DDF, DB2®, SOM, LSFM, and Internet Connection Server) with the MVS workload management (WLM) component.

This information identifies the problems that led to workload management, and describes the high-level objectives for the long term.

Problems addressed by workload management

The problems that needed to be solved by workload management include:

- ***System externals:***

Performance and tuning externals are scattered between MVS and various subsystems, as well as throughout monitoring and reporting products. This is true for the OPT parameters, and, prior to z/OS V1R3, for the MVS installation performance specifications (IPS) and the installation control specifications (ICS). MVS and the subsystems each have their own terminology for similar concepts, each have their own controls, and the controls are not coordinated.

- ***Expectations:***

Many of the MVS externals are geared towards implementation. You tell MVS how to process work, not your expectations of how well it should run work. There is no single way to make sure that important work is getting the necessary system resources.

- ***Integration and feedback:***

The multiple monitoring and reporting products show many different views of how well MVS is doing, or how well individual subsystems are managing work. Since there is no single way to specify performance goals for your installation, it is also difficult to get feedback from monitors and reporters on how well your installation actually achieved what you expected it to. There is little sense of which reports and fields relate to the externals you specified.

- ***Managing towards expectations:***

Some installations configure their systems to handle a peak load. This could result in inefficient use of expensive resources. Today's externals do not allow you to reflect your performance expectations for work. Today, a system programmer must completely understand the implications of changing a parameter before he or she can configure an installation to achieve performance objectives. Since there is no direct path between specification and expectation, it is difficult to predict the effects of changing any one control or parameter.

Given this mix of problems, workload management has some high level objectives to provide a solution.

MVS workload management solution for today and tomorrow

Workload management requires a shift of focus from tuning at a system resources level to defining performance expectations. This requires a basic shift in philosophy towards goal-oriented systems management. The complete workload management solution offers a shift in philosophy in the following areas:

- ***Fewer, simpler, and consistent system externals:***

Workload management provides a way to define MVS externals and tune MVS without having to specify low-level parameters. The focus is on setting performance goals for work, and letting the workload manager handle processing to meet the goals.

- ***Externals reflect customer expectations:***

Workload management provides new MVS performance management externals in a *service policy* that reflects goals for work, expressed in terms commonly used in service level agreements (SLA). Because the terms are similar to those commonly used in an SLA, you can communicate with end-users, with business partners, and with MVS using the same terminology.

- ***Expectations-to-feedback correlation:***

With one common terminology, workload management provides feedback to support performance reporting, analysis, and modelling tools. The feedback describes performance achievements in the same terms as those used to express goals for work.

- ***“System Managed” toward expectations:***

Workload management provides automatic work and resource management support that dynamically adapts as needed. It manages the trade-offs between meeting your service goals for work and making efficient use of system resources.

- ***Sysplex externals and management scope:***

Workload management eliminates the need to micro-manage each individual MVS image, providing a way for you to increase the number of systems in your installation without greatly increasing the necessary skill level.

Chapter 2. What is MVS workload management?

Installations today process different types of work with different response times. Every installation wants to make the best use of its resources and maintain the highest possible throughput and achieve the best possible system responsiveness.

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as CPU and storage, should be given to the work to meet its goal.

An installation should know what it expects to accomplish in the form of performance goals, as well as how important it is to the business that each performance goal be achieved. With workload management, you define performance goals for work, and the system matches resources to the work to meet those goals, constantly monitoring and adapting processing to meet the goals. Reporting reflects how well the system is doing compared to its goals.

Figure 1 shows a high level overview of the workload management philosophy. The following section explains the performance administration and performance management processes.

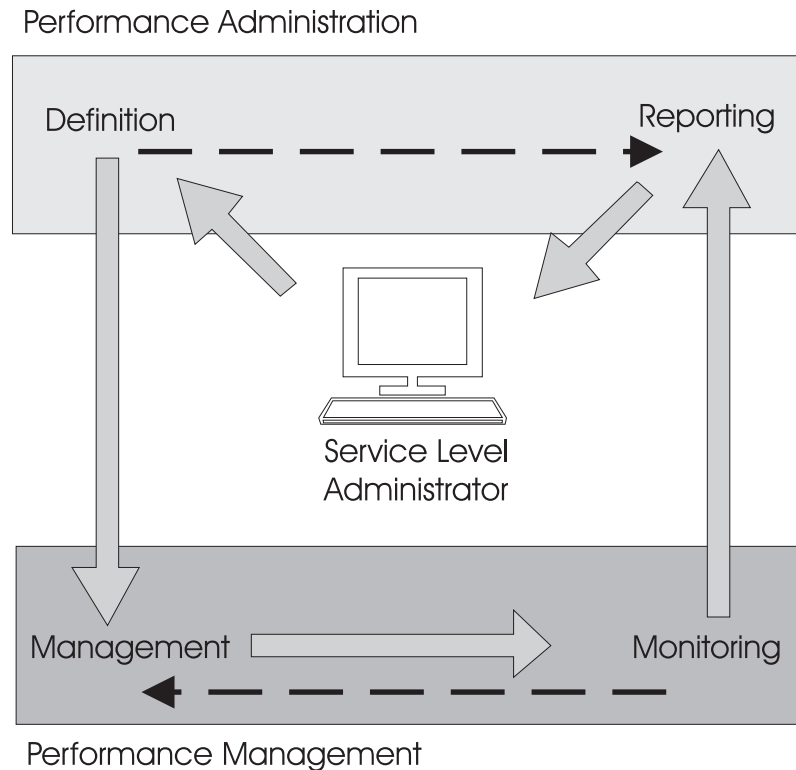


Figure 1. MVS Workload Management Overview

Performance administration

Performance administration is the process of defining and adjusting performance goals. Workload management introduces the role of the service level administrator. The service level administrator is responsible for defining the installation's performance goals based on business needs and current performance. This explicit definition of workloads and performance goals is called a *service definition*. Some installations might already have this kind of information in a service level agreement (SLA). The service definition applies to all types of work, CICS, IMS™, TSO/E, z/OS UNIX System Services, JES, APPC/MVS, LSFM, DDF, DB2, SOM, Internet Connection Server (also referred to as IWEB) and others. You can specify goals for all MVS managed work, whether it is online transactions or batch jobs. The goals defined in the service definition apply to all work in the sysplex.

Because the service definition terminology is similar to the terminology found in an SLA, the service level administrator can communicate with the installation user community, with upper level management, and with MVS using the same terminology. When the service level requirements change, the service level administrator can adjust the corresponding workload management terms, without having to convert them into low-level MVS parameters.

Workload management provides an online panel-based application for setting up and adjusting the service definition. You specify the service definition through this ISPF administrative application.

Workload management provides the capability to collect performance and delay data in context of the service definition. The performance and delay data are available to reporting and monitoring products, so that they can use the same terminology.

Performance management

Performance management is the process workload management uses to decide how to match resources to work according to performance goals. Workload management algorithms use the service definition information and internal monitoring feedback to check how well they are doing in meeting the goals. The algorithms periodically adjust the allocation of resource as the workload level changes.

For each system, workload management handles the system resources. Workload management coordinates and shares performance information across the sysplex. How well it manages one system is based on how well the other systems are also doing in meeting the goals. If there is contention for resources, workload management makes the appropriate trade-offs based on the importance of the work and how well the goals are being met.

Workload management can dynamically start and stop server address spaces to process work from application environments. Workload management starts and stops server address spaces in a single system or across the sysplex to meet the work's goals.

You can turn over management of batch initiators to workload management, allowing workload management to dynamically manage the number of batch initiators for one or more job classes to meet the performance goals of the work.

In addition to internal feedback monitoring, workload management keeps track of what is happening in the sysplex in the form of real time performance data collection, and delay monitoring. All this information is available for performance monitors and reporters for integration into detailed reports.

Workload balancing

To make the most of workload management, work needs to be properly distributed so that MVS can manage the resources. It is essential that the subsystems distributing work are configured properly for workload distribution in a sysplex. You do this with the controls provided by each subsystem. For example, in a JES2 and JES3 environment, you spread initiator address spaces across each system.

Initial cooperation between MVS and the transaction managers (CICS, IMS, DB2) allows you to define performance goals for all types of MVS-managed work. Workload management dynamically matches resources (access to the processor and storage) to work to meet the goals.

CICS, however, goes further with the CICSplex Systems Manager (CICSplex[®] SM) to dynamically route CICS transactions to meet the performance goals. CPSM monitors the performance of CICS resources and systems and presents the resources as though they were part of a single system. This type of cooperation greatly improves CICS transaction workload balancing.

Other subsystems also have automatic and dynamic work balancing in a sysplex. For example, DB2 can spread distributed data facility (DDF) work across a sysplex automatically. DB2 can also distribute work in a sysplex through its sysplex query parallelism function. CICS, TSO, and APPC cooperate with VTAM[®] and workload management in a sysplex to balance the placement of sessions. SOMobjects[®] can automatically spread its servers across a sysplex to meet performance goals and to balance the work.

For more detail on workload management in different subsystem environments, see Chapter 3, “Workload management participants,” on page 17.

Workload management concepts

The *service definition* contains all the information about the installation needed for workload management processing. There is one service definition for the entire sysplex. The service level administrator specifies the service definition through the WLM administrative application. The service level administrator sets up *policies* within the service definition to specify the goals for work. A service level administrator must understand how to organize work, and be able to assign it performance objectives.

What is a service definition?

A *service definition* consists of:

- One or more *service policies*, which are named sets of overrides to the goals in the service definition. When a policy is activated, the overrides are merged with the service definition. You can have different policies to specify goals for different times. Service policies are activated by an operator command or through the ISPF administrative application utility.

- *Service classes*, which are subdivided into periods, group work with similar performance goals, business importance, and resource requirements for management and reporting purposes. You assign performance goals to the periods within a service class.
- *Workloads*, which aggregate a set of service classes for reporting purposes.
- *Report classes*, which group work for reporting purposes. They are commonly used to provide more granular reporting for subsets of work within a single service class.
- *Resource groups*, which define processor capacity boundaries within a system or across a sysplex. You can assign a minimum and maximum amount of CPU service units on general purpose processors, per second, to work by assigning a service class to a resource group.
- *Classification rules*, which determine how to assign incoming work to a service class and report class.
- *Application environments*, which are groups of application functions that execute in server address spaces and can be requested by a client. Workload management manages the work according to the defined goal, and automatically starts and stops server address spaces as needed.
- *Scheduling environments*, which are lists of resource names along with their required states. If a z/OS image satisfies all of the requirements in a scheduling environment, then units of work associated with that scheduling environment can be assigned to that z/OS image.

The following section explains each of the concepts.

Why use service policies?

A *service policy* is a named set of overrides to the performance goals and processing capacity boundaries in the service definition. A policy applies to all of the work running in a sysplex. Because processing requirements change at different times, service level requirements may change at different times. If you have performance goals that apply to different times or a business need to limit access to processor capacity at certain times, you can define multiple policies.

In order to start workload management processing, you must define at least one service policy. You can activate only one policy at a time.

Organizing work into workloads and service classes

To workload management, work is a demand for service, such as a batch job, an APPC, CICS, DB2, or IMS transaction, a TSO/E logon, a TSO/E command, or a SOM request. All work running in the installation is divided into workloads. Your installation may already have a concept of workload. A *workload* is a group of work that is meaningful for an installation to monitor. For example, all the work created by a development group could be a workload, or all the work started by an application, or in a subsystem.

Within a workload, you group work with similar performance characteristics into *service classes*. You create a service class for a group of work that has similar:

- Performance goals
- Resource requirements
- Business importance to the installation

Performance goals

You assign a *performance goal* to each service class period, such as a response time goal, and you indicate an importance. *Importance* is how important it is to your business that the performance goal be achieved.

There are three kinds of goals:

- *Response-time goals* indicate how quickly you want your work to be processed. Since response time goals are not appropriate for all kinds of work, such as long running batch jobs, there are execution velocity goals.
- *Execution velocity goals* define how fast work should run when ready, without being delayed for processor, storage, I/O access, and queue delay. Execution velocity goals are intended for work for which response time goals are not appropriate, such as started tasks, or long running batch work.
- *Discretionary goals* are for low priority work for which you do not have any particular performance goal. Workload management then processes the work using resources not required to meet the goals of other service classes.

Resource requirements

Because some work may have variable resource requirements, you can define multiple periods for a service class. *Periods* are a way of defining different goals for work depending on the amount of resources the work consumes. Typically, periods are used to give shorter transactions more aggressive goals and to give longer running work of the same type less aggressive goals. If you have multiple periods, each period except the last has a duration. *Duration* is the amount of resources (including all processor types), in service units, that the work consumes before it is switched to the goals of the next period.

You can also group work into a service class based on resource requirements. If you have a group of batch work that can consume vast amounts of resources, and you want to limit it, you can define a service class and assign it to a resource group with a maximum amount of capacity. If the work exceeds that capacity, workload management slows the execution rate. Also, if a certain group of work needs a minimum amount of processor capacity, you can set up a service class and assign it to a resource group.

Business importance

When there is not sufficient capacity for all work in the system to meet its goals, business importance is used to determine which work should give up resources and which work should receive more. You assign an *importance* to a service class period, which indicates how important it is that the goal be met relative to other goals. Importance plays a role only when a service class period is not meeting its goal. There are five levels of importance: lowest (5), low, medium, high, and highest (1).

Figure 2 on page 8 shows an example of the relationship between workloads and service classes. Work in the figure is represented by different size triangles. The IMS workload in the figure, represents all of the IMS work. There are three single-period service classes set up, each with a different importance, and a different response time goal.

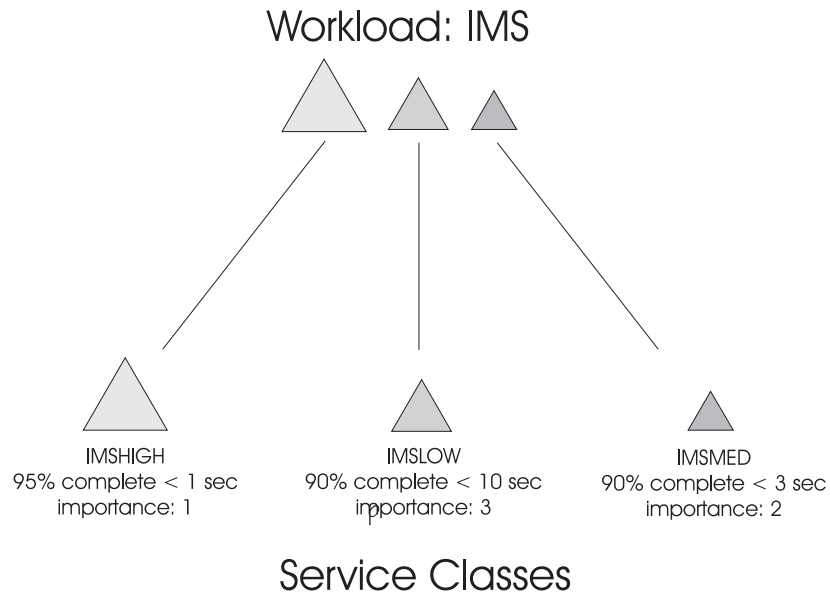


Figure 2. Workload organized by subsystem

Figure 3 shows an example of a workload that is organized by division. In the figure, work is represented by different shapes: circles, squares, and triangles. The OFFICE workload represents all of the work in the office division. There are three service classes set up, each for a different kind of work in the OFFICE workload. The IMSTEST class represents the IMS test work, CICS represents all of the CICS work, and JES represents all of the batch work in the OFFICE workload. Each of the service classes has one period with a different response time goal assigned to it.

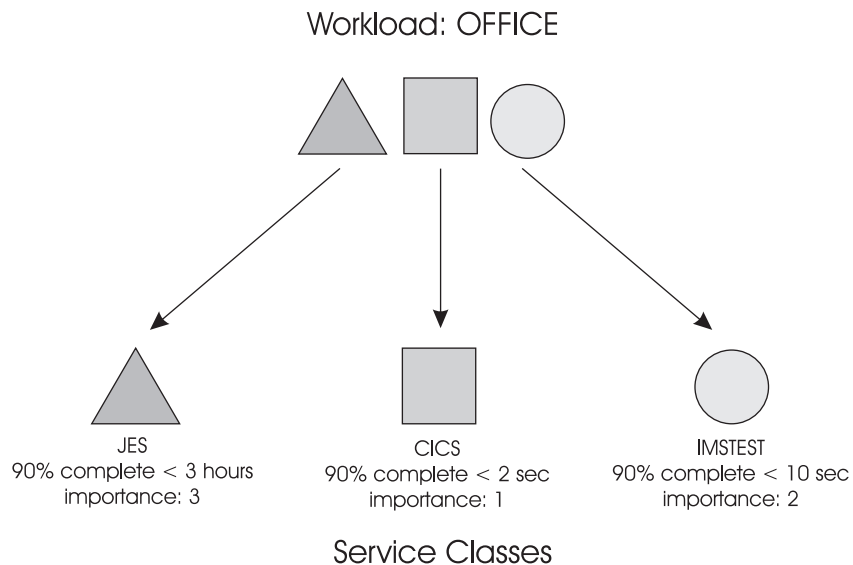


Figure 3. Workload organized by department

Why use resource groups?

Resource groups are a way of limiting or guaranteeing general purpose resource capacity. A resource group is a named amount of CPU capacity on general purpose processors that you can assign to one or more service classes. For most systems, you can let workload management decide how to manage the resources in the

sysplex and not use resource groups. You set performance goals for work and let workload management adjust to meet the goals.

In some cases, however, you might want to use a resource group, for example, to limit the service that a certain service class can consume. Note that it is recommended that each resource group is assigned to only one service class. Using a resource group is appropriate, for example:

- If you have an SLA that charges for a fixed amount of CPU capacity and you want to ensure that no more is used.
- If you want to ensure that some work with discretionary goals receives some minimum amount of capacity.

Generally, it is recommended to evaluate whether using a resource group fulfills your requirements best, or whether you let workload management take care of managing the resources.

Note: The sysplex capacity values of the resource groups apply to general purpose processors only and not to specialty processors. WLM manages resource groups based on their consumption of general purpose processor capacity.

For a resource group, you specify either a minimum or a maximum amount of general purpose processor capacity in unweighted CPU service units per second, or both.

Figure 4 shows some examples of resource groups.

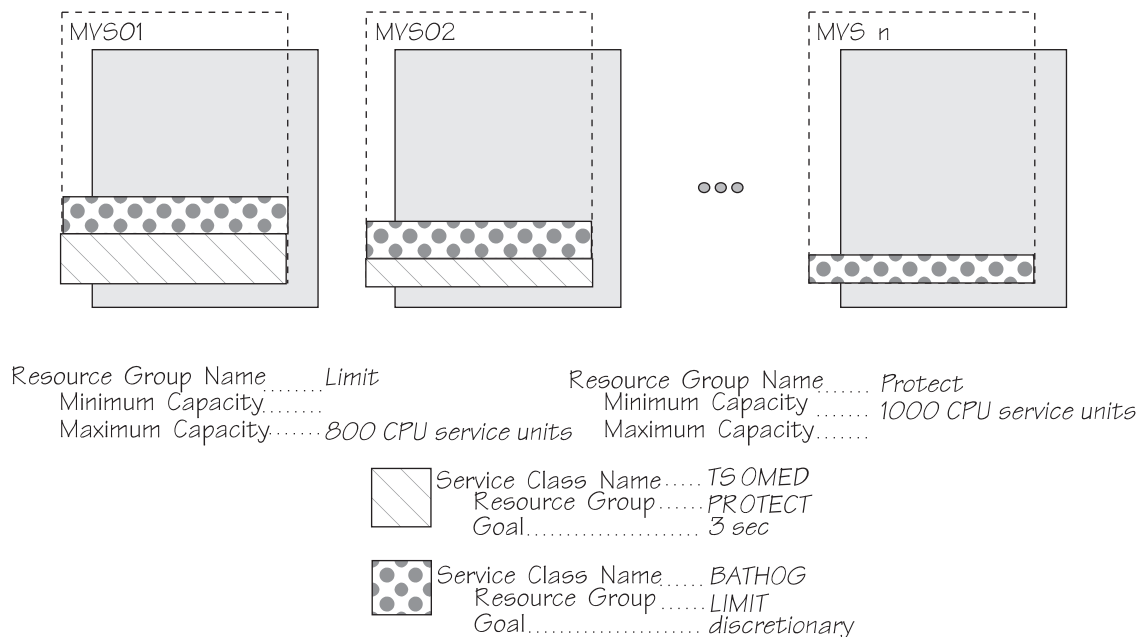


Figure 4. Resource groups

In Figure 4, the BATHOG service class is assigned to the LIMIT resource group. BATHOG might include work that consumes processing capacity in huge amounts, so you assign it to a resource group, and limit it to a maximum of 800 CPU service units per second. Also, the goal assigned to BATHOG is discretionary.

The TSOMED service class, on the other hand, is associated with the PROTECT resource group because, according to an SLA, it is contracted to be guaranteed a minimum amount of processing capacity. So the PROTECT resource group is assigned a minimum of 1000 CPU service units per second. Then, when there is sufficient TSOMED work in the system, and TSOMED is underachieving its goal, TSOMED is guaranteed at least 1000 CPU service units per second. Note that if TSOMED is surpassing its goal, then the minimum capacity setting has no effect. Workload management then processes work from both service classes, making sure that TSOMED gets its minimum amount of capacity, and making sure BATHOG does not consume more than its assigned maximum. Keep in mind the service class goal: if BATHOG is assigned a stringent goal, the goal may never be achievable within the LIMIT resource group capacity. You should determine whether the resource group capacity fulfills your purpose, or the goal. Using both controls could introduce some conflict, and the resource group controls will prevail.

Assigning work to a service class

Classification rules are the rules workload management uses to associate a performance goal and/or a resource group with work by associating incoming work with a service class. Optionally, classification rules can also associate incoming work with a report class, similar to a report performance group.

The classification rules for a subsystem are specified in terms of transaction qualifiers such as job name or transaction class. These qualifiers identify groups of transactions that should have the same performance goals and importance. The attributes of incoming work are compared to these qualifiers and, if there is a match, the rule is used to assign a service class to the work. A subsystem can also have a default class for work that does not match any of the rules.

Figure 5 on page 11 shows how work classification rules associate a service class, and optionally a report class with incoming work.

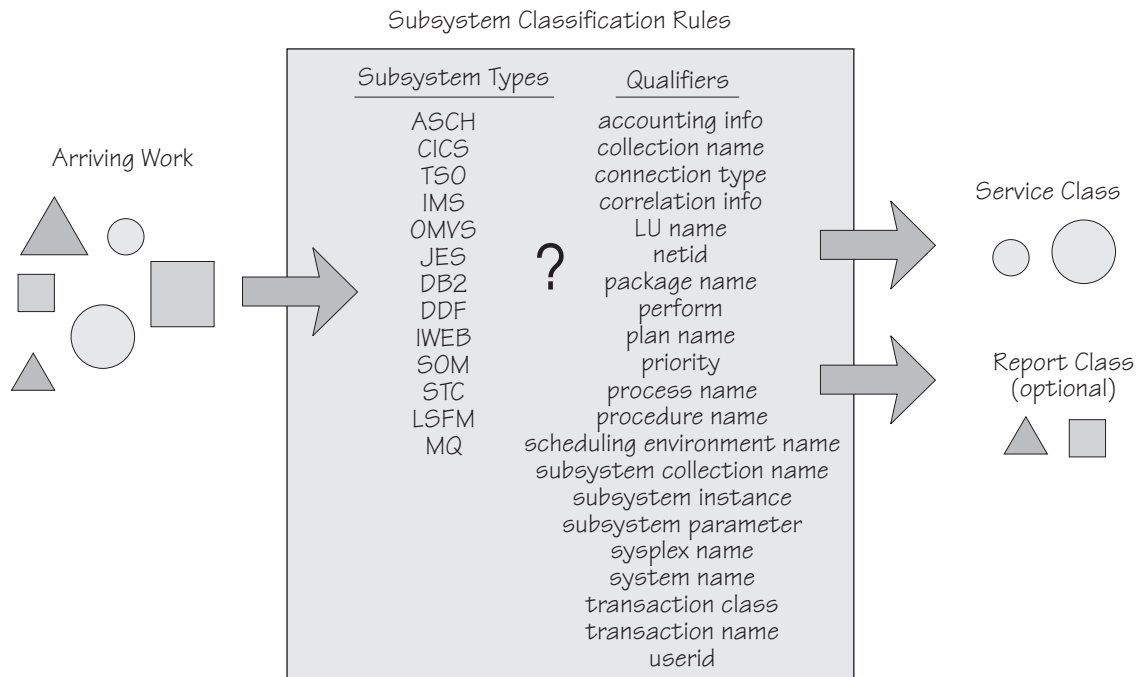


Figure 5. Work classification

Optionally, classification rules can assign incoming work to a report class. You get reporting information in terms of workloads, service classes, and report classes. Use report classes for obtaining data on a subset of work within a service class, or for rolling up data across workloads.

Why use application environments?

An *application environment* is a way to group similar server programs together and have workload management dynamically create and delete server address spaces as needed to handle the work. Each application environment typically represents a named group of server functions that require access to the same application libraries. Depending on the subsystem's implementation of application environments, the scope of server address space management is either confined to a single system or is sysplex-wide. There is also an option to manually start and stop the server address spaces for an application environment if there is a special or temporary requirement to control the number of servers independently of workload management.

If you are using a subsystem that takes advantage of application environments, you need to refer to the subsystem documentation for guidance on how to use them for that subsystem. For a list of the IBM-supplied subsystems currently using application environments, see Chapter 13, "Defining application environments," on page 125.

Figure 6 on page 12 shows an example of how two application environments, AE1 and AE2, can be used to handle work requests from a subsystem. In this example, three types of work requests, X, Y, and Z are handled. X and Y might be two different kinds of payroll inquiries, and Z might be a loan inquiry.

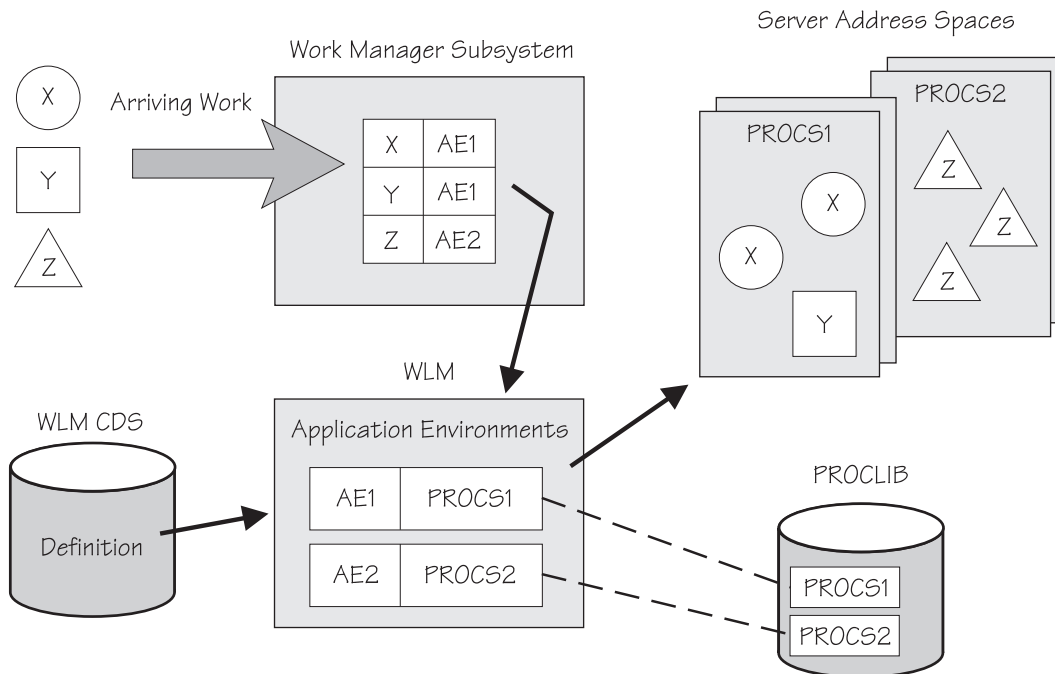


Figure 6. Application environment example

The names X, Y, and Z are used by clients when making the work requests. The work manager subsystem contains a table or file that associates the work request names with an application environment name; in this example, X and Y are associated with application environment AE1, and Z with AE2. The application environment names AE1 and AE2 are specified to workload management in the service definition stored in the WLM couple data set, and are included in the active policy when a policy is activated.

Each application environment must be assigned a system procedure library (PROCLIB) member name that contains the JCL required to start server address spaces for the application environment. In the example, PROCS1 and PROCS2 are associated with application environments AE1 and AE2, respectively.

When the work manager subsystem receives a type Y work request from a client, the subsystem requests that workload management associate the request with AE1. Workload management determines if a server address space is available to handle the request, or if an address space needs to be created. If a server address space exists, workload management makes the request available for the server. If a server address space does not exist or if more are required, workload management starts a server address space using the startup JCL procedure named PROCS1 which is defined for AE1 in the active policy.

Workload management dynamically creates new server address spaces if they are needed to handle more incoming work and, for certain subsystems such as DB2, decreases the number of server address spaces if less capacity is needed. Refer to Chapter 13, "Defining application environments," on page 125 for a description of how to define application environments to workload management.

Why use scheduling environments?

Scheduling environments help ensure that units of work are sent to systems that have the appropriate resources to handle them. A scheduling environment is a list

of resource names along with their required states. Resources can represent actual physical entities, such as a data base or a peripheral device, or they can represent intangible qualities such as a certain period of time (like second shift or weekend).

These resources are listed in the scheduling environment according to whether they must be set on or off. A unit of work can be assigned to a specific system only when all of the required states are satisfied. This function is commonly referred to as *resource affinity scheduling*.

Figure 7 shows a simple scheduling environment example. The arriving work units, X, Y, and Z, could be batch jobs submitted through JES2 or JES3. Each of the jobs had a scheduling environment associated with it at the time it was submitted (in this case the X and Y jobs are each associated with the A scheduling environment, and the Z job is associated with the B scheduling environment).

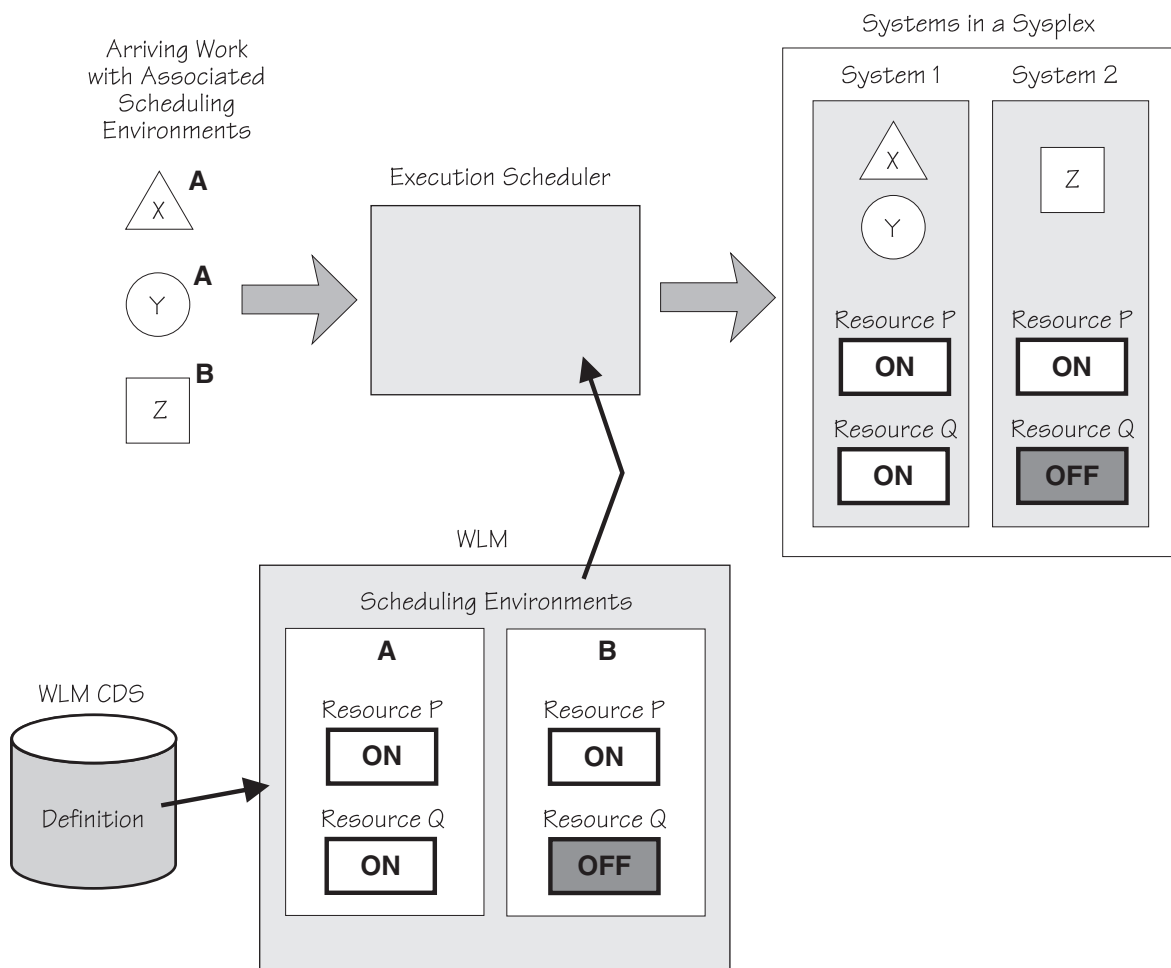


Figure 7. Scheduling environment example

JES checks the scheduling environment associated with each arriving batch job and then assigns the work to a system that matches that scheduling environment. In the example, both the X and Y jobs require that both Resource P and Resource Q be set to ON, so those jobs can be initiated only on System 1 in the sysplex. The Z job requires that Resource P be set to ON and that Resource Q be set to OFF. So that job can be initiated only on System 2.

In a sysplex containing only one system, scheduling environments have some degree of usefulness, as JES will hold batch jobs until the required states become satisfied. In a multisystem sysplex, the full power of scheduling environments becomes apparent, as work is assigned only to those systems that have the correct resource states (the resource affinity) to handle that work.

Presently, JES2 and JES3 are the only participants that use scheduling environments, although the concepts could certainly apply to other types of work in the future.

See Chapter 14, "Defining scheduling environments," on page 137 for a description of how to define scheduling environments to workload management.

Summary of service definition and service policy concepts

When you set up your service definition, you identify the workloads, the resource groups, the service classes, the service class periods, and goals based on your performance objectives. Then you define classification rules and one or more service policies. This information makes up the base service definition.

A two-step process is required before the sysplex starts using a new service definition. First, you install the service definition onto the WLM couple data set. Second, you activate one of the service policies from the definition.

With a service policy, you can override specific goals or resource groups in your base service definition. In a typical scenario, you might define a base service definition that is appropriate for your normal business hours. Because you need to have at least one service policy defined, you might create an empty service policy called NORMAL. While the NORMAL service policy is in effect, there would be no overrides to the goals or resource groups in the base service definition. If you have a special business need to change your goals for offshift processing, you might then also create a service policy called OFFSHIFT. If you were to activate this policy at the end of the business day (either by invoking the VARY WLM,POLICY=*poli*cyname command or by using the "Activating a Service Policy" panel in the ISPF application), then the goal overrides in the OFFSHIFT service policy would be in effect until you switched back to NORMAL the next morning.

Chapter 5, "Defining service policies," on page 35 tells you more about how to define a policy, and also shows a few examples.

Note that you can override *only* goals, number and duration of periods, resource group assignments and values. All of the workloads, service class names, classification rules, scheduling environments, and application environments defined in the service definition remain the same for any policy. If you need to change any of these, you will need to change the base service definition, reinstall the service definition, and then activate a policy from that changed service definition.

Note, also, that you need to define all of your policies at the outset, while you are defining the rest of the service definition. Once the service definition is installed, then you can switch from one defined policy to another. If you need to create a new policy or change the overrides in an existing policy, you will need to reinstall the service definition with the new or changed policy definition before you can activate the new policy.

Figure 8 shows a service definition with two service policies.

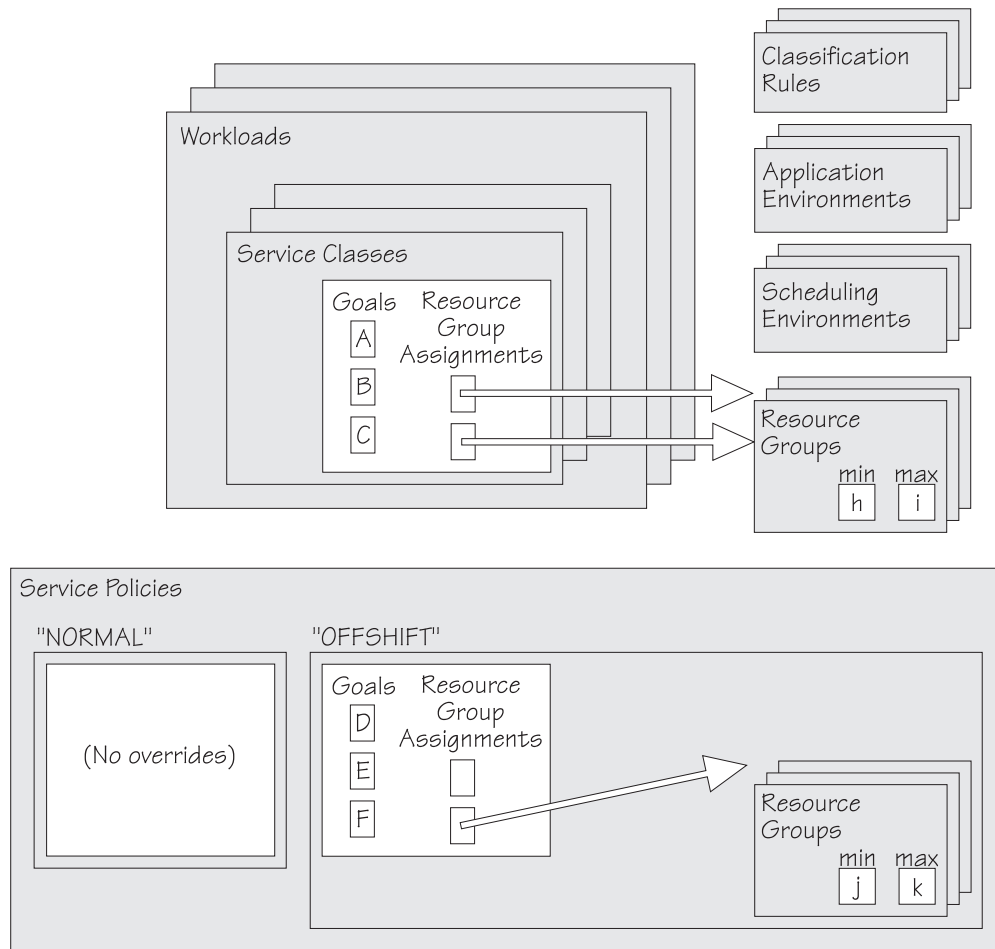


Figure 8. Service definition, including two service policies. The NORMAL service policy is empty; therefore, the originally defined goals, resource group assignments, and resource group attributes remain unchanged when the NORMAL service policy is in effect. When the OFFSHIFT service is in effect, certain goals, resource group assignments, and resource group attributes are overridden.

Chapter 3. Workload management participants

You use the WLM ISPF administrative application to define your service definition. The administrative application requires the following products:

- TSO/E Version 2.5 plus SPEs, or later
- ISPF 4.3, or later

This information describes the work and reporting environments that support workload management.

Workload management work environments

Cooperation between MVS and the subsystem work managers enable the sysplex-wide specification and reporting of goals defined for work in the service policy. You can define goals for the following kinds of work:

- IMS, if you have IMS/ESA Release 5 or higher
- CICS, if you have CICS/ESA 4.1 or higher
- z/OS UNIX System Services
- JES2
- JES3
- APPC
- TSO/E
- SOMobjects
- WebSphere® Application Server objects (CB)
- LSFM
- DDF, if you have DB2 V4.1, or higher
- DB2, if you have DB2 V 5, or higher
- IWEB, if you have Internet Connection Server V 2.2, or higher, Domino® Go Webserver, or IBM http Server Powered by Domino (IHS powered by Domino)
- MQSeries® Workflow
- SYSH, if you need to manage non z/OS partitions
- NETV

Arriving work is associated with a service class, and therefore a goal and possibly a resource group. You get feedback from RMF™ as to how well the system is doing in meeting the goals. RMF provides sysplex-wide workload management reporting, providing sysplex-wide as well as single system feedback on the goals through the Post Processor and Monitor III realtime reports. The reports show kinds of delays seen by subsystem work managers such as CICS and IMS. SDSF displays workload management related information.

The Workload Management Environment

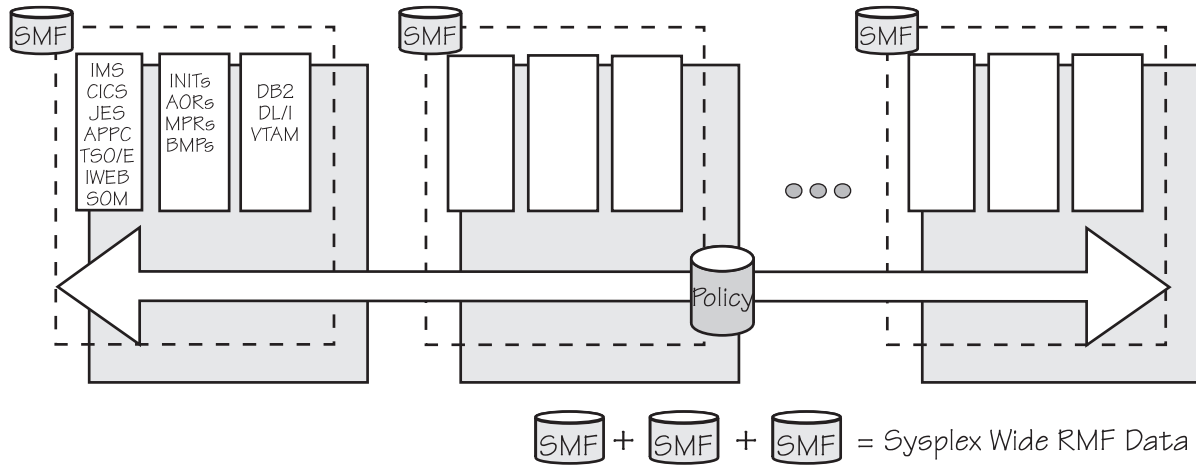


Figure 9. Sysplex view of the management environment. The shaded box represents the processor(s) and the dashed box represents the MVS image(s) running on the processor.

Workload management understands which address spaces (INITs, AORs, MPPs, BMPs, TORs, FORs) are involved in processing work within a service class, and matches the resources to meet the goal. Information as to how well each system is doing in processing towards a service class goal is recorded in SMF records on each system. For more information about workload management information in RMF reports, see *z/OS RMF Report Analysis*.

Subsystem support for goal types and multiple periods

The types of goals a subsystem supports depends on the workload management services it uses. For the following subsystems that have address space-oriented transactions or use enclaves, you can specify any goal type and multiple periods:

- Subsystems that have address space-oriented transactions:
 - APPC
 - JES2
 - JES3
 - z/OS UNIX System Services
 - TSO/E
- Subsystems that use enclaves:
 - WebSphere Application Server
 - DB2
 - DDF
 - IWEB
 - MQSeries Workflow
 - LSFM
 - NETV
 - SOMobjects

Note: Enclaves are transactions that can span multiple dispatchable units in one or more address spaces, and in the case of multisystem enclaves, one or more address spaces on multiple systems in a parallel sysplex. See *z/OS MVS Programming: Workload Management Services* for more information about enclaves.

The CICS and IMS subsystems do not use enclaves, but use a different set of WLM services to support their transactions to WLM. Therefore, they support only response time goals, either percentile or average, and single period service classes.

Subsystem-specific performance hints

Based on installation experiences, here are some subsystem-specific performance hints:

- Watch out for increased CPU usage by the WLM address space due to a high CICS **MAXTASK** setting.

For CICS 4.1 and higher releases, WLM collects delay samples for each performance block. Because the number of performance blocks created is based on the **MAXTASK** value (a value of 100 means 100 performance blocks created per region), a **MAXTASK** value that is too high can cause a large sampling overhead when a CICS workload is switched to goal mode. If **MAXTASK** has been set to an arbitrarily high value, it should be reduced to a true “high water mark” value.

- Watch out for work defaulting to SYSOTHER.

Work in subsystems that use enclaves (see “Subsystem support for goal types and multiple periods” on page 18 for a list of these subsystems) can suffer performance degradation if left unclassified in the service definition. If you do not add classification rules for this work in your service definition, it will be assigned to the SYSOTHER service class, which has a discretionary goal. Using the WLM application, you need to add classification rules to assign the work to service classes with appropriate response time or velocity goals.

As a general rule, it's a good idea to keep an eye on the SYSOTHER service class through RMF or another monitor. Any service accumulating in the SYSOTHER service class is a signal that you have unclassified work in your system.

For the latest information on these topics and others, see the WLM/SRM website at:

<http://www.ibm.com/servers/eserver/zseries/zos/wlm/>

Workload balancing

Workload management allocates resources to meet goals of the work that arrives. System programmers must use the existing methods of routing and scheduling work for subsystems except for those listed. For subsystems not exploiting workload balancing or routing services, if you want to balance your work across all MVS images in a sysplex, the system programmer must set the routing controls to either balance the arrival of work, or to ensure that all MVS images are equal candidates for processing work.

Examples of subsystems that can automatically balance work in a sysplex include:

- CPSM provides goal-oriented routing based on the goal defined for CICS work in the workload management service policy.
- DB2 V 4.1 provides automatic and dynamic work balancing in a sysplex for distributed data facility (DDF) work.
- DB2 V 5 provides additional automatic work balancing through its sysplex query parallelism function.
- SOMobjects uses application environments to help balance object class binding requests in a sysplex.
- CICS V 4.1, DB2 V 4.1, TSO/E V 2.5, and APPC cooperate with VTAM 4.4 and workload management in a sysplex to manage session placement. New sessions

for these subsystems are directed to the appropriate systems in the sysplex to balance work and meet performance goals.

- JES2 and JES3 provides automatic and dynamic placement of initiators for WLM-managed job classes. z/OS 1.4 together with JES2 1.4, and z/OS 1.5 together with JES3 1.5 provide Initiator Balancing, so that already available WLM managed initiators can be reduced on fully loaded systems and increased on low loaded systems to improve overall batch work performance and throughput over the sysplex.
- WebSphere Application Server cooperates with WLM in a sysplex to balance work among application control regions and to meet performance goals.

Workload management in a CPSM environment

Figure 10 shows workload management in a CICS with CPSM environment.

WLM recognizes that the terminal owning region (TOR) and the application owning region (AOR) on one or more systems are involved in processing CICS transactions. Using RMF Monitor I, you can get reporting information on the CICS response times, and on any execution delays experienced by a service class period for a single system or for the sysplex.

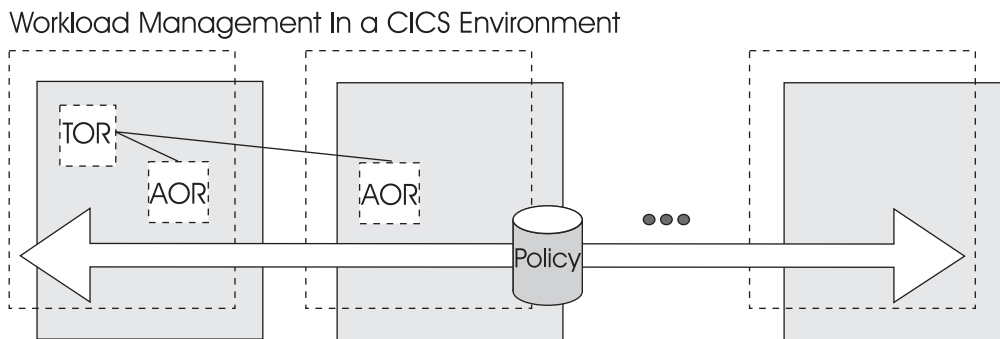


Figure 10. Workload management in a CICS environment

In a CPSM environment, WLM provides the CICS service class goal to CPSM. If the goal is an average response time goal and you specified the dynamic goal-oriented algorithm for CPSM, then CPSM uses the transaction's goal to help decide where to route the transaction for processing to meet the goal. If the goal is a percentile response time goal, CPSM reverts to its "shortest queue" algorithm since it only has average response time data available to it. Percentile goals are still preferred for any workload that can have a few unusually long transactions distorting the average response time. Following is a summary of the two CPSM algorithms:

Shortest Queue

Send the CICS transaction to the AOR that has the shortest queue of pending work, but prefer AORs on a local MVS image, and be sure to avoid "unhealthy" AORs.

Goal Send the transaction to an AOR that has been successfully meeting the goal, but prefer AORs on a local image and be sure to avoid "unhealthy" AORs. If the goal is a percentile goal, use the shortest queue.

For more information, see *CICSplex SM Managing Workloads*.

Workload management in a DB2 Distributed Data Facility environment

The definition of a response time for the enclaves used to manage DB2 Distributed Data Facility (DDF) transactions depends upon several parameters, including DB2 installation parameters and the attributes used when binding the package or plan. For more information, see *DB2 Administration Guide*.

Batch workload management

Workload management can dynamically manage the number of batch initiator address spaces in a JES2 or JES3 environment. You can selectively turn over control of the batch initiator management to WLM for one or more job classes. WLM will start new initiators, as needed, to meet the performance goals of this work.

By specifying or defaulting `MODE=JES` on the JES2 `JOBCLASS` statement or the JES3 `GROUP` statement, you indicate that the initiators for the job class should be JES-managed, as in the past. By specifying `MODE=JES`, you keep the job class in JES-managed mode. (JES will manage the batch initiators for that job class, in the same way it has in prior releases.) By specifying `MODE=WLM`, you put that class into WLM-managed mode.

You can switch as many job classes to WLM-managed mode as you wish. You can easily switch any job class back to JES-managed mode by using the JES2 `$TJOBCLASS` command or the JES3 `MODIFY` command.

Note:

1. If you have velocity performance goals set for the work running on WLM-managed batch initiators, be aware that the initiation delays will be figured into the velocity formula. This will affect your velocity values and probably require some adjustment to your goals. See Chapter 8, “Defining service classes and performance goals,” on page 49 for information on defining velocity goals.
2. All jobs with the same service class should be managed by the same type of initiation. For example, if jobs in job classes A and B are classified to the `HOTBATCH` service class, and `JOBCLASS(A)` is `MODE=WLM`, while `JOBCLASS(B)` is `MODE=JES`, workload management will have a very difficult time managing the goals of the `HOTBATCH` service class without managing class B jobs.
3. You can use the JES2 `JOBCLASS` parameter to specify a default scheduling environment, thereby saving the effort of changing JCL jobcards or writing a specific JES2 exit to assign the scheduling environment.
4. If a WLM initiator is experiencing `ABEND822s`, there are two ways to recycle the initiator:
 - If you can determine the ASID of the initiator that is abending, you can stop it by issuing the `P INIT,A=asid` command. The initiator does not need to be idle at the time that you enter the command. If the initiator is busy processing a job, it will stop itself after the job finishes. WLM will automatically replace the initiator with a new one.
 - If you cannot determine the ASID, or if you want to recycle all initiators as part of a regular cleanup process, you can enter the `$P XEQ` and `$S XEQ` commands. The `$P XEQ` command causes all WLM initiators on that system to be “flagged” to terminate. The `$S XEQ` command enables WLM to start new initiators (without needing to wait for the old initiators to end). Beware that the `$P XEQ` command purges WLM's history which tells it how many

initiators are needed for each service class. It may take some time for WLM to build up the same number of initiators that existed before.

See the following JES2 and JES3 documentation for more information about WLM-managed JES2 job classes:

- *z/OS JES2 Initialization and Tuning Guide*
- *z/OS JES2 Initialization and Tuning Reference*
- *z/OS JES3 Initialization and Tuning Guide*
- *z/OS JES3 Initialization and Tuning Reference*
- *z/OS JES3 Commands*

The following other functions exist to help you manage batch work in a JES environment:

- A new work qualifier, **PRI**, which allows you to use the job priority when defining work classification rules. See Chapter 9, “Defining classification rules,” on page 61.
- Scheduling environments, which allow you to define resource requirements for incoming work, ensuring that the work will be scheduled on a system within the sysplex only if the resource settings on that system satisfy those requirements. See Chapter 14, “Defining scheduling environments,” on page 137.

Multisystem enclave support

With multisystem enclave support, enclaves can run in multiple address spaces spanning multiple systems within a parallel sysplex. As in a single system enclave, the work will be reported on and managed as a single unit.

z/OS UNIX System Services Parallel Environment uses multisystem enclaves to run parallel jobs. With all tasks of the job running in the same enclave, WLM can manage all of the work to a single performance goal.

See Chapter 17, “Defining a coupling facility structure for multisystem enclave support,” on page 169 for more information on setting up the SYSZWLM_WORKUNIT coupling facility structure, a prerequisite to multisystem enclave support.

See the “Creating and Using Enclaves” topic in *z/OS MVS Programming: Workload Management Services* for more information on multisystem enclaves.

See *z/OS UNIX System Services Parallel Environment Operation and Use* for more information on UNIX System Services Parallel Environment.

Intelligent Resource Director

The Intelligent Resource Director (IRD) extends the concept of goal-oriented resource management by allowing you to group logical partitions that are resident on the same physical server, and in the same sysplex, into an “LPAR cluster.” This gives WLM the ability to manage resources, both processor and DASD I/O, not just in one single image but across the entire cluster of logical partitions.

See Chapter 18, “The Intelligent Resource Director,” on page 173 for more information.

HiperDispatch mode

This information briefly describes the HiperDispatch mode.

Overview

In addition to the performance improvements available with the IBM System z10™ processors, z/OS workload management and dispatching are enhanced to take advantage of the System z10 hardware design. A mode of dispatching called HiperDispatch provides additional processing efficiencies.

The HiperDispatch mode aligns work to a smaller subset of processors to maximize the benefits of the processor cache structures, and thereby, reduce the amount of CPU time required to execute work. Access to processors has changed with this mode, and as a result, LPAR weights prioritization of workloads via WLM policy definitions becomes more important.

The concept of HiperDispatch mode

Without HiperDispatch, for all levels of z/OS, a TCB or SRB may be dispatched on any logical processor of the type required (standard, zAAP or zIIP). A unit of work starts on one logical processor and subsequently may be dispatched on any other logical processor. The logical processors for one LPAR image will receive an equal share for equal access to the physical processors under PR/SM™ LPAR control. For example, if the weight of a logical partition with four logical processors results in a share of two physical processors, or 200%, the LPAR hypervisor will manage each of the four logical processors with a 50% share of a physical processor. All logical processors will be used if there is work available, and they typically have similar processing utilizations.

With HiperDispatch mode, work can be managed across fewer logical processors. A concept of maintaining a working set of processors required to handle the workload is introduced. In the previous example of a logical partition with a 200% processor share and four logical processors, two logical processors are sufficient to obtain the two physical processors worth of capacity specified by the weight; the other two logical processors allow the partition to access capacity available from other partitions with insufficient workload to consume their share. z/OS limits the number of active logical processors to the number needed based on partition weight settings, workload demand and available capacity. z/OS also takes into account the processor topology when dispatching work, and it works with enhanced PR/SM microcode to build a strong affinity between logical processors and physical processors in the processor configuration.

Processor categories

The logical processors for a partition in HiperDispatch mode fall into one of the following categories:

- Some of the logical processors for a partition may receive a 100% processor share, meaning this logical processor receives an LPAR target for 100% share of a physical processor. This is viewed as having a high processor share. Typically, if a partition is large enough, most of the logical partition's share will be allocated among logical processors with a 100% share. PR/SM LPAR establishes a strong affinity between the logical processor and a physical processor, and these processors provide optimal efficiencies in HiperDispatch mode.
- Other logical processors may have a medium amount of physical processor share. The logical processors would have a processor share greater than 0% and up to 100%. These medium logical processors have the remainder of the

partition's shares after the allocation of the logical processors with the high share. LPAR reserves at least a 50% physical processor share for the medium processor assignments, assuming the logical partition is entitled to at least that amount of service.

- Some logical processors may have a low amount, or 0%, of physical processor share. These “discretionary” logical processors are not needed to allow the partition to consume the physical processor resource associated with its weight. These logical processors may be parked. In a parked state, discretionary processors do not dispatch work; they are in a long term wait state. These logical processors are parked when they are not needed to handle the partition's workload (not enough load) or are not useful because physical capacity does not exist for PR/SM to dispatch (no time available from other logical partitions).

When a partition wants to consume more CPU than is guaranteed by its share and other partitions are not consuming their full guaranteed share, a parked processor can be unparked to start dispatching additional work into the available CPU cycles not being used by other partitions. An unparked discretionary processor may assist work running on the same processor type.

When examining an RMF CPU activity report in HiperDispatch mode, one may now see very different processing utilizations across different logical processors of a logical partition. For further information, refer to *z/OS RMF Report Analysis*.

Setting HiperDispatch mode in SYS1.PARMLIB

The HiperDispatch state of the system is determined by the number of logical processors defined on an LPAR and the `HIPERDISPATCH=YES|NO` keyword in the `IEAOPTxx` member of `SYS1.PARMLIB`.

All partitions with more than 64 logical processors defined at IPL are forced to run with `HIPERDISPATCH=YES`. LPARs with more than 64 logical processors defined are also unable to switch into `HIPERDISPATCH=NO` after IPL.

For all partitions with less than 64 logical processors, HiperDispatch is enabled or disabled by the `HIPERDISPATCH=YES|NO` keyword in parmlib member `IEAOPTxx`. This parameter can be changed dynamically with the use of the **SET OPT** command. This enables the operating system to choose the desired mode of operation.

When a new hardware generation is installed, for any z/OS image(s) that are running with HiperDispatch disabled, the system programmer should reevaluate whether those z/OS image(s) should be migrated to running with HiperDispatch enabled in the new environment.

On z10 on any z/OS release, HiperDispatch disabled is the default. However, customers are encouraged to run with HiperDispatch enabled on z10 to take advantage of the processing benefits.

Beginning with z/OS V1R13 on IBM zEnterprise® 196 (z196), HiperDispatch enabled is the default. With z/OS V1R13 running on a z196, z/OS partitions with share greater than 1.5 physical processors will typically experience improved processor efficiency with HiperDispatch enabled. z/OS partitions with share less than 1.5 physical processors typically do not receive a detectable performance improvement with HiperDispatch enabled, but IBM recommends running those LPARs with HiperDispatch enabled when the performance improvement is greater than or equal to HiperDispatch disabled.

There are no new hardware controls or settings to enable use of HiperDispatch within a logical partition; however, the existing “control global performance data” security setting must be enabled on HMC for proper operation of HiperDispatch in a logical partition. HiperDispatch cannot effectively utilize vertical low processors when other partitions are active on the system and “global performance data” is not enabled.

For further information about the **HIPERDISPATCH** parameter, refer to *z/OS MVS Initialization and Tuning Reference*.

I/O storage management

Workload management can pass service class importance and goal information to the storage I/O priority manager in the IBM System Storage® DS8000® series. The information enables the storage I/O priority manager to provide favored processing for I/O requests of important z/OS workloads that are not achieving their goals.

The storage I/O priority manager may throttle I/O requests to facilitate favored access to storage server resources for other I/O requests. The storage I/O priority manager analyzes the properties of the service class period associated with an I/O request and determines whether the I/O request should be favored, or throttled.

Handling service class periods with a response time goal

For service class periods with a response time goal, the goal achievement and specified importance are analyzed. Service class periods that exceed their goal may be throttled if there are service class periods that do not achieve their goal. Service class periods that miss their goal might only be throttled if there are service class periods with a higher importance that do not achieve their goal. In detail, service class periods are processed as follows:

- Service class periods with a response time goal and importance 1 that do not achieve their goal are favored. They are not throttled when they exceed their goal.
- Service class periods with a response time goal and importance 2 that do not achieve their goal are favored. They are not throttled when they exceed their goal, except when more important work is missing its goals.
- Service class periods with a response time goal and importance 3 that do not achieve their goal are favored. They might be moderately throttled when they exceed their goal or more important work is missing its goals.
- Service class periods with a response time goal and importance 4 that do not achieve their goal are moderately favored. They might be throttled when they exceed their goal or more important work is missing its goals.
- Service class periods with a response time goal and importance 5 may be throttled when they exceed their goal or more important work is missing its goals. They are not favored when they miss their goal.

The level of favored processing or throttling that the I/O requests of a service class period with a certain importance receive, depends on the goal achievement, that is, the performance index (PI), of the service class period. A service class period does not receive favored processing if its performance index is between 0.9 and 1.4. The more the PI of a service class period exceeds 1.4, the more it is favored: moderate, strong, or very strong. The more the PI of a service class period falls below 0.9, the more it may be throttled: moderate, strong, or very strong.

Table 1 shows the possible levels of favored processing, or throttling in relationship to the importance and goal achievement of a service class period with response time goal:

Table 1. Levels of processing

Importance	Service class period misses goal (PI is higher than or equals 1.4)	Service class period exceeds goal (PI is lower than or equals 0.9)
1	Favored processing — moderate, strong, or very strong	Not throttled
2	Favored processing — moderate, strong, or very strong	Not throttled, except when there are service class periods with higher importance that miss their goal
3	Favored processing — moderate or strong	Throttled — moderate
4	Favored processing — moderate	Throttled — moderate or strong
5	No favored processing	Throttled — moderate, strong, or very strong

Handling service class periods with a velocity goal

For service class periods with a velocity goal, the specified velocity goal and importance are taken into account.

Service class periods with a high importance (1 or 2) and a high velocity goal are most likely being favored. Service class periods with a low importance and a low velocity goal might be throttled.

Because the goal achievement of service class periods with a velocity goal is not taken into account, I/O requests of these service class periods might be throttled even when they miss their goal and might be favored even when they exceed their goal.

Handling other I/O requests

I/O requests associated with the system-provided service classes SYSTEM, SYSSTC, or SYSSTC1 - SYSSTC5 are not managed by the I/O priority manager.

I/O requests associated with service class periods that have a discretionary goal may be throttled, but will never be favored.

Controlling the information passed to the I/O manager

The STORAGESEVERMGT=YES|NO parameter in the IEAOPTxx member of SYS1.PARMLIB controls whether service class importance and goal information is passed to the storage I/O priority manager.

STORAGESEVERMGT=YES specifies that SRM should provide service class importance and goal information to the storage I/O priority manager. The default is STORAGESEVERMGT=NO. Before specifying STORAGESEVERMGT=YES, verify that your IBM System Storage DS8000 model incorporates the I/O priority manager feature. Furthermore, verify if the service option I/O Priority Management in the WLM service definition is set to YES.

Throttle delays introduced by the storage I/O priority manager are reflected in control unit queue delays. Therefore, if STORAGESEVERMGT=YES is specified, control unit queue delays are not considered when the execution velocity is calculated for service class periods with a velocity goal.

If you have significant control unit queue delays in your installation, you might have to adjust the velocity goal of service class periods when you specify `STORAGESERVERMGT=YES`.

For further information about the `STORAGESERVERMGT` parameter, refer to *z/OS MVS Initialization and Tuning Reference*.

Non-z/OS partition CPU management

If you have logical partitions running Linux or other non-z/OS systems, you can manage CPU resources across these logical partitions in accordance with workload goals. Non-z/OS partition CPU management does not support the management of partitions running z/OS systems.

Non-z/OS partition CPU management allows WLM to shift weight from either z/OS or non-z/OS partitions to another z/OS or non z/OS partition in the same LPAR cluster.

A new SYSH subsystem type is introduced for Linux CPU management. SY and PX are the qualifiers which are valid for SYSH. No resource groups can be associated with service classes for SYSH. Also CPU protection cannot be assigned. SYSH supports velocity goals with single periods, but no discretionary goals.

In order to activate non-z/OS CPU Management, see “Enabling non-z/OS CPU management” on page 180.

Workload management and Workload License Charges

As part of the z/OS support of Workload License Charges, you can set a defined capacity limit, also called a *soft cap*, for the work running in a logical partition. This defined capacity limit is measured in millions of service units (MSUs) per hour. It allows for short-term spikes in the CPU usage, while managing to an overall, long-term *four-hour rolling average* utilization. It applies to all work running in the partition, regardless of the number of individual workloads the partition may contain.

See *PR/SM Planning Guide* for more information about how to set a defined capacity limit.

WLM enforces the defined capacity limit by tracking the partition's CPU usage and continually averaging it over the past four hours. Spikes higher than the defined capacity limit are possible, as long as they are offset by low points that keep the four-hour average at or below the limit. When this four-hour average goes over the defined capacity limit, then WLM caps the partition. At that point, it can use no more than the defined capacity limit, until the average drops below the limit.

At IPL, WLM defaults to a four-hour time interval that contains no partition CPU usage. This means the four hour rolling average starts with zero.

Consider the example shown in Figure 11 on page 28:

Defined capacity limit enforcement

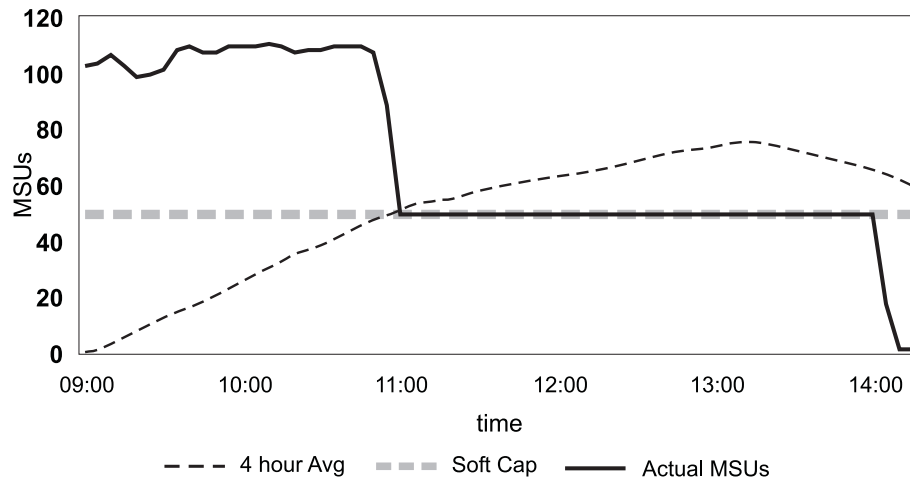


Figure 11. Workload management enforcing a defined capacity limit

This partition has a 50 MSU defined capacity limit, as shown by the grey dashed line. The solid black line is the actual MSU consumption of the partition, and the dashed black line is the four-hour rolling average utilization. At IPL (09:00), the partition's CPU usage starts with more than 100 MSUs. The four-hour average, however, is below the defined capacity limit at that point, due to the fact that WLM is using the default four-hour interval time containing no partition CPU usage. No action is taken. Just before 11:00, the four-hour average hits the defined capacity limit. Now WLM caps the partition's CPU usage at 50 MSUs.

WLM allows the CPU usage to remain at the defined capacity limit. Therefore, the four-hour average can continue to go up. In this example, this occurs because the low usage numbers of the 4-hour default time interval at IPL are falling off the back end of the four-hour horizon, and being replaced by the new usage numbers starting at IPL (09:00). This is a consequence of managing to the four-hour average.

In summary, WLM caps a partition's CPU usage only when the four-hour average reaches the defined capacity limit. Before that point, you may see CPU usage spikes over the limit. You may see the four-hour average go over the limit after capping, until the passage of time gradually brings the average back down to the limit.

Defining the capacity of a partition

WLM instructs the PR/SM hypervisor to cap the partition, as described in the previous example, in one of three possible ways that depend on the ratio of the defined capacity limit and the weight of the partition. The weight defined for a partition determines the capacity share of the partition within the shared processor capacity. The following cap mechanisms will be used when the four-hour rolling average exceeds the defined capacity limit:

- **Option 1:**
If the capacity share exactly equals the defined capacity limit, WLM instructs PR/SM to fully cap the partition at its weight.
- **Option 2:**

If the capacity share is less than the defined capacity limit it is not possible to permanently cap the partition at its weight because the partition would not be able to use the capacity it is entitled to. WLM defines a cap pattern which caps the partition at its weight for some amount of time over an interval and then removes the cap during the remaining time of the interval. On average this appears as if the partition was constantly capped at its defined capacity limit. The cap pattern depends on the ratio of the capacity share to the defined capacity limit. If the capacity share based on the weight is rather small compared to the defined capacity limit the partition will be capped very drastically during short periods of time. Therefore, for configurations requiring the capacity share to be smaller than the defined capacity limit, it is recommended to keep both definitions as close as possible.

- **Option 3:**

If the capacity based on the weight is greater than the defined capacity limit, WLM instructs PR/SM hypervisor to define a phantom weight, because it is not possible for WLM to cap the partition at its capacity based on the weight. A phantom weight pretends a utilization of the CPC for a particular partition which makes it possible to cap the partition to be managed at the defined capacity limit.

Options 1 and 3 are the recommended ways of specifying defined capacity limits and weights because they provide a capping behavior which is very smooth.

zEC12 (GA2) and later systems support the use of negative phantom weights in Option 3. For systems running on eligible hardware with the required software support (z/OS V2R1) option 2 above is obsolete and option 3 will be used. Consequently the smoother flavor of capping will always be used regardless of the ration of the capacity share to the defined capacity limit.

Defining group capacity

Group capacity limit is an extension of the defined capacity limit. It allows an installation to define a “soft cap” for multiple logical partitions of the same CPC (all running z/OS V1R8 or later). The group limit is a defined capacity (soft cap) for all partitions defined in the group. The capacity group is defined on the Hardware Management Console (HMC). Each capacity group has a name and a defined capacity which becomes effective to all partitions in the group.

See *PR/SM Planning Guide* for more information about how to define a capacity group.

WLM uses the weight definitions of the partitions and their actual demand to decide how much capacity may be consumed by each partition in the group. In the following example a capacity group is defined which consists of three partitions MVS1, MVS2 and MVS3. The group limit is defined to 50 MSU and the weights of the partitions are shown in Table 2:

Table 2. Example of definitions for MVS1, MVS2, and MVS3 partitions

Partition	Weight	Share (MSU)
MVS1	100	16,7
MVS2	50	8,3
MVS3	150	25

zEC12 (GA2) and later systems support the use of the initial weight for sharing the group limit. If all partitions of the capacity group are running on this hardware and if all of them also have the required software support installed (z/OS V2R1, z/OS V1R12/13 with OA41125), the initial weight will be used instead of the current weight to calculate the share of the partitions.

The total weight of all partitions in the group is 300. Based on the weight definitions, each partition gets an entitled share of the group capacity of 50 MSU. The entitled share is important to decide how much MSU can be used by each partition if the 4-hour rolling average of the group exceeds the group capacity limit. The share is also shown in Table 2 on page 29.

Figure 12 on page 31 shows an example how the partitions use their entitled capacity. At 07:00 p.m. all three partitions are started. In the beginning, only partition MVS1 and MVS2 use approximately 60 MSU. No work is running on partition MVS3. Therefore its measured consumption is very small. WLM starts to cap the partitions when the 4-hour rolling average for the combined usage exceeds the 50 MSU limit. This happens around 09:00 p.m. At that point, MVS1 is reduced to about 30 MSU and MVS2 to about 20 MSU. MVS3 still does not demand much CPU. Therefore the available MSU of the group can be consumed by MVS1 and MVS2.

Around 11:00 p.m. work is started on MVS3. A small spike can be observed when WLM recognizes that the third partition starts to demand its share of the group. After that spike MVS3 gets up to 25 MSU of the group because its weight is half of the group weight. MVS1 is reduced to 16.7 MSU and MVS2 to 8.3 MSU. Based on variation in the workload the actual consumption of the partitions can vary but the group limit of 50 MSU is always met on average.

The work on MVS3 stops around 04:00 p.m. At that point, a small negative spike can be observed and afterwards the capacity is consumed again only by the partitions MVS1 and MVS2.

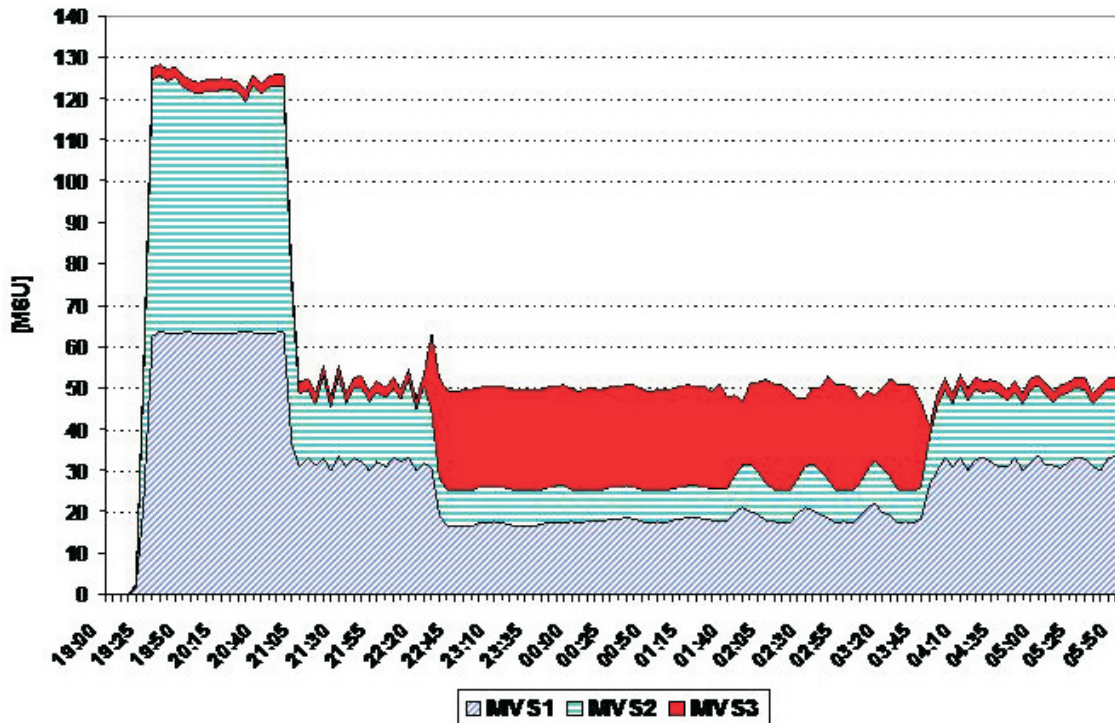


Figure 12. Example of workload consumption for partitions MVS1, MVS2, and MVS3

Group capacity can be combined with all other existing management capabilities of the z/OS Workload Manager:

- Group capacity can be combined with an individually defined capacity for a partition. The partition will always honor the individually defined capacity.
- It is possible to define multiple capacity groups on a CPC. A partition can only belong to one capacity group at a given point in time.

Note that WLM only manages partitions which comply with the following rules within a group:

- The partitions must not be defined with dedicated processors.
- The partitions must run with shared CPs and wait completion must be No.
- The operating system must be z/OS V1R8, or higher.
- A hardware cap (that is, **Initial Capping** checked on the Change Logical Partitions Controls panel) which limits the partition to its weight is not allowed for a partition being managed in a capacity group.

All partitions which do not conform to these rules are not considered part of the group. WLM will dynamically remove such partitions from the group and manage the remaining partitions towards the group limit.

Group capacity functions together with IRD weight management while the partitions in the group are not subject to capping. No weight moves take place for partitions as long as the group is being capped.

Group capacity does not work when z/OS is running as a z/VM® guest system.

Workload management with other products

For subsystem work managers, if you do not have the product or product release supporting workload management, you can define goals only for the subsystem regions. Since subsystem regions are treated as batch jobs or started tasks, a velocity goal is most appropriate. All started tasks are managed on the system where they are started.

For example, if you have a CICS release that is not at least at the CICS/ESA 4.1 level, you could define velocity goals for the TORS and AORS. You could not manage CICS transactions to response time goals.

For reporting or monitoring products, you should check whether they support workload management. Vendor information should explain whether or not they report on the workload management activity.

Chapter 4. Setting up a service definition

This information describes how to set up a service definition based on your performance objectives. A service definition contains all of the information necessary for workload management processing. Based on this information, you should be able to set up a preliminary service definition using the worksheets provided. Then, you can enter your service definition into the ISPF administrative application with ease.

The service definition is the way to express your installation's business goals to your sysplex. In order to do this, you must understand your installation's business environment from the following areas:

- What are your installation's revenue-earning workloads?
- Is there a business priority to the workloads?
- Do you understand what kind of service users expect and the timeframe they expect it in?
- Do you understand the service you can deliver?
Even if your installation does not currently have an SLA, or other written performance objectives, users most often have service expectations.
- How can you use monitoring and performance products to gather information?

Specifying a service definition

You define a minimum of one policy in a service definition. A service definition contains workloads, service classes, and classification rules. Optionally, it contains resource groups, report classes, and application environments. A service definition also includes one or more policies. A policy is a set of overrides to the goals and resource group limits in the service definition. In a service definition, you also specify whether you want workload management to manage I/O priorities based on your service class goals. A service definition has an identifying name and description.

There is one notepad available for the service definition. You can update the notepad whenever you are creating or modifying any of the other parts of a service definition, such as the policy, workload, service class, resource group, and classification rules. All of the information in a service definition is available to reporting products, except for the notepad information and the classification rules. The notepad information is provided for a history log, and change management.

Name Service definition name

Description

Description of the service definition

Name

(Required) An eight-character identifier of the service definition.

Description

(Optional) An area of 32 characters to describe the service definition. For example, you could include the time period this service definition is intended to cover.

Storing service definitions

You can work in the ISPF administrative application with one service definition at a time. In order to make the service definition accessible to all systems in the sysplex, you store the service definition on a WLM couple data set. Only one service definition can be installed on the WLM couple data set at a time.

If you want to work on more than one service definition at a time, you can keep each in a distinct MVS partitioned data set (PDS), or in an MVS sequential data set (PS). As an MVS service definition data set, the service definition is subject to all the same functions as an MVS data set. You can restrict access to the service definition data set, send it, and copy it, as you can any MVS data set.

The service definition must be installed on a WLM couple data set, and a service policy activated. Only service policies in the service definition installed on the WLM couple data set can be activated. A WLM couple data set can have automatic backup. For more information about allocating and using a WLM couple data set, see Chapter 16, "Workload management migration," on page 153.

Defining the parts of a service definition

This information explains how to set up a service definition by defining each of its parts: policies, workloads, resource groups, application environments, service classes, classification rules, and report classes. When you set up your service definition, you should define its parts in the following order:

1. Service policy
See Chapter 5, "Defining service policies," on page 35.
2. Workloads
See Chapter 6, "Defining workloads," on page 39.
3. Resource groups
See Chapter 7, "Defining resource groups," on page 41.
4. Service classes
See Chapter 8, "Defining service classes and performance goals," on page 49.
5. Service policy overrides
See "Using policy overrides" on page 36.
6. Classification rules
See Chapter 9, "Defining classification rules," on page 61.
7. Report classes
See Chapter 10, "Defining report classes," on page 99.
8. Service coefficients and options
See Chapter 11, "Defining service coefficients and options," on page 101.
9. Application environments
See Chapter 13, "Defining application environments," on page 125.
10. Scheduling environments
See Chapter 14, "Defining scheduling environments," on page 137.

Chapter 5. Defining service policies

A *service policy* is a named collection of service class and resource group specification overrides. When a policy is put into effect, the overrides are merged with the service class and resource group specifications in the service definition. A policy override is a way to change a goal or resource group capacity without having to redefine all of your service classes and resource groups.

See “Summary of service definition and service policy concepts” on page 14 for an overview of the relationship between a service definition and a service policy, and Figure 8 on page 15 for a visual overview of how service policy overrides work. Note that in an ideal scenario, you would only have to define your service definition once. As part of that service definition, you would predefine multiple policies to meet varying performance goals or business needs. Once the service definition is installed, you would then activate one policy at a time, and then, when appropriate, switch to another. Note that you *must* define at least one service policy, and you can define up to 99.

When you are creating your service definition, you may choose to define one empty “default” policy with no overrides at all. Next, create your workloads and service classes. Then determine how and when your service classes may have different goals at different times. Define additional policies with the appropriate overrides for these time periods.

Name Service policy name

Description

Description of the service policy

Policy override

Changing a service class goal or resource group

Name

(Required) Eight characters identifying the service policy. Every service policy name must be unique in a service definition. The service policy is activated by name in one of the following ways:

- An operator command from the operator console.
- A utility function from the workload management ISPF application.

You can display the name of the active service policy with an operator command, or by viewing a performance monitor, such as RMF.

Description

(Optional) An area of 32 characters describing the service policy. The descriptive text is available to performance monitors for reporting.

Policy override

(Optional) A way to change a performance goal, a service class-resource group assignment, or a resource group capacity for a service policy. For more information about defining policy overrides, see “Using policy overrides” on page 36.

Examples of service policies

- Daytime policy:

Name = DAYTIME

Description = Policy from 9:00 am to 5:00 pm

- Policy for national holidays:
Name = HOLIDAY
Description = Policy for Arbor day
- Weekend policy:
Name = WEEKEND
Description = Policy for Sat and Sun

Using policy overrides

Once you have defined your service classes, you can determine whether any of your service class goals change at different times. If they do, you can define a policy override. With an override, you can change one or more of the following for a service policy:

- A goal for a service class period
- Number and duration of periods
- A service class - resource group assignment
- Resource group attributes
- CPU protection.

Example 1: Policy overrides

In this example, the service class BATPIG is in the BATCH workload. It is associated with the resource group LIMIT. Suppose the LIMIT resource group is assigned some maximum capacity. BATPIG is assigned a discretionary goal. Since it is a discretionary goal, it does not have an assigned importance. In the Weekend policy, however, both the goal and the resource group association is overridden. The resource group association is overridden, so that in the Weekend policy, BATPIG is not assigned to a resource group. It is instead assigned a response time goal of 1 hour, with an importance of 5.

```

----- Base -----
Policy: Standard                               Policy: Weekend
                                           (overrides)

Service Class..... BATPIG
Description..... All batch CPU hogs
Workload..... BATCH
Resource Group. LIMIT

Period   Goal.... discretionary           1 hour AVG
1        Import.. n/a                     5

```

Example 2: Policy overrides

In this example, the CICSHIGH service class is in the CICS workload. In the Standard policy, it is assigned a response time goal of 1 second average, with an importance of 1. It is not assigned a resource group, because there is no business need to limit or guarantee capacity. For the weekend policy, however, it has an overridden goal. Because of a contract with CICS users, the agreed response time for weekends is 2 seconds average, with an importance of 2.

```

----- Base -----
Policy: Standard                               Policy: Weekend
                                           (overrides)

Service Class..... CICSHIGH
Description..... All short CICS transactions
Workload..... CICS
Resource Group. _____

```

Period	Goal.... 1 sec avg	2 sec avg
1	Import.. 1	2

Example 3: Policy overrides

In this example, the resource group DEPT58 is associated with 3 service classes: 58CICS, 58TSO, and 58BATCH. Since the department is willing to pay for more capacity on the weekends, the minimum is overridden for the weekend policy. It is 1500 CPU service units per second. So in the weekend policy, the service classes 58CICS, 58TSO, and 58 BATCH have a minimum of 1500 CPU service units guaranteed.

```

----- Base -----
      Policy: Standard                      Policy: Weekend
                                           (overrides)

Resource Group..... DEPT58
Description..... Contracted capacity for dept. 58

Minimum      1000                          1500
Maximum      _____

Service Class 58CICS_
              58TSO_
              58BATCH_

```

Chapter 6. Defining workloads

A *workload* is a named collection of work to be reported as a unit. You can arrange workloads by subsystem (CICS, IMS), by major application (production, batch, office) or by line of business (ATM, inventory, department). Logically, a workload is a collection of service classes.

Name Workload name

Description

Description of workload

Name

(Required) Eight characters identifying the workload. Every workload name must be unique for all defined workloads in a service definition.

Description

(Optional) An area of 32 characters describing the workload. The descriptive text is available to performance monitors for reporting.

Defining a departmental workload

In order to set up a departmental workload that crosses subsystem boundaries, you must keep in mind how you can assign the work to service classes. You should review Chapter 9, “Defining classification rules,” on page 61 and find out the best way you can assign your work to service classes. The following list provides some examples of what would be required for some of the subsystems:

TSO You must set up TSO user IDs or account numbers according to department structure, so that the user IDs correspond to a specific department.

JES You must have unique batch classes or account numbers by department.

CICS You must have unique CICS regions for each department.

IMS You must have a separate IMS/VS resource lock manager (IRLM), IMS control region, and IMS message processing region (MPR) for each workload.

For more information, see Chapter 9, “Defining classification rules,” on page 61.

Examples of workloads

- By subsystem:

Name IMS

Description

All work in classes IMS1 IMSA IMSS TIMS IMSV

- By department/location:

Name DEVELOP

Description

All work in classes STSO IMSB and LINKA

Chapter 7. Defining resource groups

A *resource group* is an amount of processor capacity and memory. It is optional. Unless you have some special need to limit or protect processor capacity or memory for a group of work, you should skip defining resource groups and let workload management manage all of the processor and memory resource to meet performance goals. You use a resource group to:

- Limit the amount of general purpose processor capacity available to one or more service classes.
- Set a minimum of general purpose processor capacity for one or more service classes in the event that the work is not achieving its goals.
- Define a minimum and maximum amount of processor capacity sysplex-wide, or on system-level.
- Limit the amount of real memory capacity available to one or more service classes on system-level.

Note: The sysplex processor capacity values of the resource groups apply to general purpose processors only and not to specialty processors. WLM manages resource groups based on their consumption of general purpose processor capacity.

You can specify a minimum and maximum amount of processor capacity and a maximum amount of memory to a resource group. You can assign only one resource group to a service class in one WLM policy. You can assign multiple service classes to the same resource group. You can define up to 32 resource groups per service definition.

Keep in mind your service class goals when you assign a service class to a resource group. Given the combination of the goals, the importance level, and the resource capacity, some goals may not be achievable when capacity is restricted.

Setting a maximum processing capacity

If work in a resource group is consuming more processor resources than the specified maximum processor capacity, the system caps the associated work accordingly to slow down the rate of processor resource consumption. The system may use several mechanisms to slow down the rate of processor resource consumption, including swapping the address spaces, changing their dispatching priority, and capping the amount of processor service that can be consumed. Reporting information reflects that the service class may not be achieving its goals because of the resource group capping.

Setting a minimum processing capacity

By setting a minimum processing capacity, you create an overriding mechanism to circumvent the normal rules of importance. If the work in a resource group is not meeting its goals, then workload management will attempt to provide the defined minimum amount of CPU resource to that resource group.

Setting a memory limit

By specifying a memory limit, you explicitly restrict physical memory consumption of work that is running in address spaces that are associated with the resource

group through classification. For a resource group with a memory limit, the system creates a memory pool. The memory pool does not reserve physical memory exclusively for members of the pool, but rather keeps track of the aggregate usage by members in order to limit their total usage.

An address space that is associated with the resource group through classification connects to the memory pool. In that case, all its physical frames are counted toward the pool's limit. When a memory pool approaches its limit, the system takes actions such as initiating self-stealing to page out memory pool pages and thus free up memory pool frames. This protects the physical memory allocation of other work that is running on the system. IBM recommends that you use memory pools only when it is required to limit memory consumption for workloads such as Apache SPARK that provide guidance on how to operate them in a memory pool.

Reclassification of active address spaces to another resource group memory pool is not supported. When you reset an active address space to another service class that is associated with a different resource group with memory limit, the new memory limit is ignored, and the address space remains in its original memory pool. Unless an active address space is no longer classified to any resource group memory pool, it remains in the memory pool that it was classified to when it started its current execution, and is reclassified when it is subsequently executed again. For example, a WLM policy activation resulting in an active address space being classified to no resource group, or to a resource group with no memory pool, results in the address space being removed from the memory pool.

Shrinking the size of a memory pool is not supported. When you activate a policy with a lower memory limit for an existing resource group that is in use by some address spaces, the new memory limit is ignored. The associated memory pool keeps its original size.

When a memory pool approaches its limit, address spaces starting up and connecting to the pool are deferred until enough frames are available through self-stealing from the pool.

An active address space cannot be reclassified to another defined memory pool. The address space must terminate in order to be reclassified, or it must be reclassified to the global pool and then reclassified again to the new pool.

A resource group's memory pool limit can be dynamically increased by activating a new policy. However, a resource group's memory pool limit cannot be reduced while it is defined in a policy. To reduce the limit, you must activate a policy that does not have the resource group defined, and then activate a policy that has the resource group defined with a larger smaller limit.

Name Resource Group name

Description

Description of resource group

Resource Group Type

Description of resource group type

Capacity Maximum

Can be calculated in various ways, depending on which resource group is used, and will be explained in the following.

Capacity Minimum

Can be calculated in various ways, depending on which resource group is used, and will be explained in the following.

Memory Limit

Maximum amount of memory that address spaces associated with the resource group through classification may consume on the local system. The value has a system scope.

Name

Eight characters that identify the name of the resource group. Each resource group must be unique within a service definition.

Description

Up to 32 characters that describe the resource group.

Resource Group Type

Resource groups allow to define a guaranteed maximum and minimum CPU consumption for work on the sysplex and on each individual member of the sysplex. This allows to:

- Prioritize work on a system-level basis
- Control the minimum and maximum resource consumption

The following types of resource groups are valid:

Resource Group Type 1

The capacity is specified in unweighted CPU service units per second, the value must be between 0 and 99999999. Only the service units used on general purpose processors controls the management of the resource group.

Minimum and maximum capacity applies sysplex-wide, that is, WLM ensures that the limits are met within the sysplex.

The table in Appendix B, “CPU capacity table,” on page 255 shows the service units per second by CPU model.

Resource Group Type 2

The capacity is specified as a percentage of the LPAR share in the general purpose processor pool, the value must be between 0 and 99, and the sum of all minimum LPAR share percentages for all resource groups of this type should not exceed 99.

Minimum and maximum capacity has a system scope, that is, WLM ensures that the limits are met on each system within the sysplex. Refer to “Calculating an LPAR share — Example 1” on page 45 for a scenario showing how to calculate an LPAR share when using resource group type 2.

Resource Group Type 3

The capacity is specified as a number of general purpose processors (CPs), a number of 100 represents the capacity of 1 CP. The number should be between 0 and 999999.

Minimum and maximum capacity has a system scope, that is WLM ensures that the limits are met on each system within the sysplex.

Capacity

Identifies the amount of available capacity you want workload management to allocate to the resource group. Capacity includes cycles in both TCB and SRB

mode. The table in Appendix B, "CPU capacity table," on page 255 shows the service units per second by CPU model. Resource group minimum can equal resource group maximum.

Maximum

CPU service that this resource group may use. *Maximum* specified for this resource group applies to all service classes in that resource group combined. *Maximum* is enforced. There is no default maximum value. If specified, *Maximum* must be greater than 0.

Minimum

CPU service that should be available for this resource group when work in the group is missing its goals. The default is 0. If a resource group is not meeting its minimum capacity and work in that resource group is missing its goal, workload management will attempt to give CPU resource to that work, even if the action causes more important work (outside the resource group) to miss its goal. If there is discretionary work in a resource group that is not meeting its minimum capacity, WLM will attempt to give the discretionary work more CPU resource if that action does not cause other work to miss its goal.

The minimum capacity setting has no effect when work in a resource group is meeting its goals.

Memory Limit

Maximum amount of memory that address spaces that are associated with the resource group through classification may consume on the local system. The attribute is specified in GB. It may exceed the real storage available to a system. The attribute value has system scope.

Note:

1. You cannot assign a resource group to service classes representing transaction-oriented work, such as CICS or IMS transactions. The ISPF application notifies you with an error message if you attempt to do so. If you want to assign a minimum or a maximum processor capacity and a maximum amount of memory to CICS or IMS work, you can do so by assigning a resource group to their regions. For example, suppose you have three service classes representing your CICS works: CICSTRN, CICSAORS, and CICSTORS. CICSTRN represents all of your online CICS transactions, and has one period with a short response time goal. CICSAORS and CICSTORS represent all of your CICS AOR and TOR regions, respectively, that process the online transactions. To assign a maximum processor capacity and a maximum amount of memory to your CICSTRN work, define a resource group, and assign it to the regions. So you assign the resource group to the CICSAORS and CICSTORS service classes.
2. Similarly, resource groups with a memory limit cannot be applied to enclave service classes. However, because enclave service classes can be used anywhere, unlike CICS or IMS transaction service classes, the ISPF application does not notify you with an error message if you attempt to do so. As for CICS or IMS, a resource group with a memory limit must be assigned to the service class of the address spaces that will join the enclaves.
3. A memory limit overrules the storage critical attribute assigned in classification rules and also any protective storage target managed through SRM.
4. Resource group processor capacity capping is implemented by marking the work units that belong to resource group non-dispatchable for some time slices and dispatchable for the remaining time slices (awake slices). Depending on the configuration, it may not be possible to enforce very low resource group limits.

The granularity to which a resource group limit can be managed depends on how much service the work can consume in a system or across the Sysplex, respectively, during one awake time slice. Beginning with z/OS V2.1 the granularity of awake slices is 1/256 of the time.

- While resource groups are managed based on the general purpose processor service, the dispatchability attribute is also honored by zAAP and zIIP processors.

Calculating an LPAR share — Example 1

The following example illustrates how the capping works for a resource group type 2 and how to calculate this. For this resource group the minimum and maximum capacity is defined as a percentage of the share for the logical partition:

LPAR share

The LPAR share is defined as the percentage of the weight definition for the logical partition to the sum of all weights for all active partitions on the CEC.

In this example a resource group type 2 ELPMAX is defined to cap CPU-intensive work on the systems in the sysplex environment. The aim is to limit the consumption of the work to 60% of the LPAR share. The sysplex consists of 2 systems: WLM1 and WLM2 which share the CEC with other VM and MVS systems:

Table 3. Example: LPAR configuration

Partition	Current weight	Share	Logical processors	Sysplex
WLM1	78	7,8%	2	WLMPLEX
WLM2	132	13.2%	2	WLMPLEX
VMA	590	59.0%	6	n/a
MVSA	200	20.0%	4	n/a

The CEC is a zSeries 990, Model 306. For the value service units per second (SU/s) of 18626.3, refer to Appendix B, “CPU capacity table,” on page 255. With 6 processors the total SU/s is 111758. This value is called CEC-capacity. Based on the CEC-capacity value, it is possible to calculate the LPAR share capacity, which is the CEC-capacity multiplied with the LPAR share of each system in the sysplex. For system WLM1 the LPAR share is 7.8%. The LPAR share capacity for WLM1 results to 8717 SU/s. For ELPMAX a maximum limit of 60% is defined. On WLM1 this results to 5230 SU/s.

Table 4. Example: ELPMAX in sysplex WLMPLEX

CEC-Capacity for zSeries 990, Model 306 => 18626.3 SU/s * 6 = 111758 SU/s					
Partition	LPAR share	Share capacity (SU/s)	RG limit: 60%	Logical processors (LCPs)	SU/s based on LCPs
WLM1	7.8%	8717	5230 SU/s	2	37252.6
WLM2	13.2%	14752	8851 SU/s	2	37252.6

Note that ELPMAX is still defined sysplex-wide but the capacity definition depends on the logical and physical configuration of the systems. On WLM1 work

in ELPMAX is entitled to consume 5230 SU/s and on WLM2 to 8851 SU/s. Furthermore, the following two factors are also important for the entitled capacity:

- The SU/s based on the logical processor configuration of the LPAR. If this SU/s for the logical processor configuration is smaller than the share of the logical partition, we will calculate the entitlement for ELPMAX based on the logical processor configuration. In our example there are 2 logical processors defined per partition. This results to 37252.6 SU/s and this value is bigger than the SU/s based on the LPAR share.
- If there is a defined capacity and the partition is capped to that defined capacity, the entitlement for ELPMAX is based on that defined capacity. In the current example it is assumed that there is no defined capacity.

Figure 13 shows how work is capped in resource group ELPMAX on system WLM1. For this reason we associate a service class with ELPMAX and submit work which consumes a lot of CPU service:

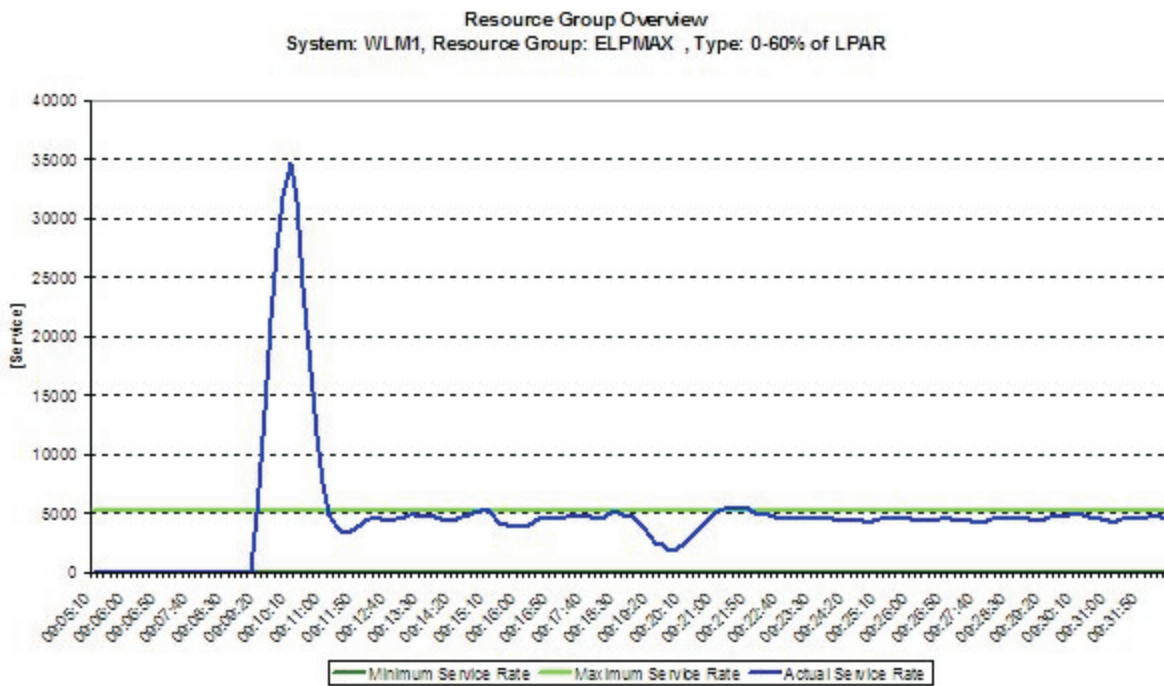


Figure 13. Example: Resource group overview

It can be observed that the system needs a short period to apply the correct number of capping slices for the work units associated with the resource group. After this ramp up period, the resource group is capped slightly below the entitled capacity. The reason for that is, that the work is being capped by slices which depend on the capacity of the partition. Therefore, the capacity used can never match the entitlement exactly.

The following shows the complete calculations for applying and interpreting the results of a resource group type 2:

$$RG_{[SU_s]} = \begin{cases} \text{Softcap}_{[SU_s]} \bullet RG_{\text{Limit}[\%]} & \text{if softcap} < \text{ShareCapacity or LCPCapacity} \\ \text{LCPCapacity}_{[SU_s]} \bullet RG_{\text{Limit}[\%]} & \text{if LCPCapacity} < \text{ShareCapacity or softcap} \\ \text{ShareCapacity}_{[SU_s]} \bullet RG_{\text{Limit}[\%]} & \text{if ShareCapacity} < \text{LCPCapacity or softcap} \end{cases}$$

$$\text{ShareCapacity}_{[SU_s]} = \text{CECCapacity}_{[SU_s]} \bullet \text{LPARShare}$$

CECCapacity_[SU_s] = Capacity based on shared physical processors for the CEC

$$\text{LPARShare} = \frac{\text{Weight}(\text{Current Partition})}{\sum_i^{\text{all active partitions}} \text{Weight}(i)}$$

LCPCapacity_[SU_s] = Capacity based on shared processors available on the LPAR

Figure 14. Working With A Resource Type 2 - Sample Calculation

RG[SU_s] is the result entitlement for the resource group which depends on the LPAR share, LCP capacity and whether there is a defined capacity which is active.

Specifying the capacity as a number of CPs — Example 2

For resource group type 3, the capacity is defined as a number of general purpose processors (CPs). The following example illustrates how the capping works for a resource group type 3 and how to calculate this. This means, finding out to how many service units per second (SU/s) the defined capacity corresponds.

The CEC is a zSeries 990, Model 316. The WLM service definition specifies a resource group RGT3 (= type 3) with a maximum/minimum value of 250. For resource group type 3 a number of 100 represents the capacity of 1 CP. The LPAR has 6 processors assigned on the hardware console and none of them have been varied offline from the MVS console. To calculate the capping value, do the following:

1. For the correct SU/s value for the resource group, refer to Appendix B, "CPU capacity table," on page 255. In the table for z990s, find the row that represents the model with the number of online processors that your LPAR has. In the current example, this is Model 306, because your LPAR has 6 online processors.

Note: The higher number in this row compared to the value for Model 316 accounts for the lower MP factor that your LPAR has because it has only 6 processors.

2. So, the value you pick is 18 626.3.
3. Multiply 18 626.3 by 250/100.
4. The result is: on that system, the maximum capacity for RGT3 is 46 565 SU/s.

The calculation is done individually for each LPAR. If the LPARs in the sysplex have a different number of processors assigned to them - one can still calculate the correct capping value. This means, the same resource group may represent different SU/s on different LPARs.

An advantage of using resource group type 3 is that it dynamically adjusts to the processor capacity when the work is run on another hardware.

Chapter 8. Defining service classes and performance goals

A *service class* is a named group of work within a workload with the following similar performance characteristics:

- Performance goals
- Resource requirements
- Business importance to the installation

Workload management manages a service class period as a single entity when allocating resources to meet performance goals. A service class can be associated with only one workload. You can define up to 100 service classes.

You can assign the following kinds of *performance goals* to service classes: average response time, response time with percentile, velocity, and discretionary. You assign an importance level to the performance goal. Importance indicates how vital it is to the installation that the performance goal be met relative to other goals.

Because some work has variable resource requirements, workload management provides *performance periods* where you specify a series of varying goals and importances. You can define up to eight performance periods for each service class. You can also assign a service class to a resource group if its CPU service must be either protected or limited.

This information explains the parts of a service class, how to define performance goals, and how to use performance periods.

Defining service classes and performance goals	
Name	Service class name
Description	Service class description
Workload	The name of the workload associated with this service class.
Resource Group	The name of the resource group associated with the work in this service class.
Performance Period	One goal per period.
Duration	Number of service units for this performance period. This value is calculated including all processor types.
Average Response Time	Average response time for transactions completing within the period in terms of hours, minutes, and seconds. Decimal points are accepted. Response time varies from 15 milliseconds to 24 hours.
Response Time and Percentile	A percentile of work to be completed in the specified amount of time. Percentile boundaries vary from 1 to 99. Amount of time is in hours, minutes, or seconds. Decimal points are accepted. Response time ranges from 15 milliseconds to 24 hours.
Velocity	Measure of how fast work should run when ready, without being delayed for WLM-managed resources. Velocity ranges from 1 to 99.
Discretionary	Workload management defined goal. Work is run as system resources are available.
Importance	How important it is to the installation that the goal be achieved.
CPU Protection	Whether long-term CPU protection should be assigned to this service class.
I/O Priority Group	Whether long-term I/O protection should be assigned to this service class.
Honor Priority	Whether this service class is exempted from the system-wide IFAHONORPRIORITY or IIPHONORPRIORITY processing as specified in parmlib member IEAOPTxx

Name (required)

Eight characters describing the service class. Service class names must be unique within a service definition.

Description (optional)

An area of 32 characters describing the service class. The descriptive text is available to performance monitors for reporting.

Workload (required)

The name of the workload associated with the service class. You can associate only one workload per service class in a service definition. The workload must have been previously defined.

Resource Group (optional)

The resource group name associated with this service class. You can assign only one resource group per service class in a service policy. You can override the resource group assigned to a service class in each service policy. For more information about resource groups, see Chapter 7, “Defining resource groups,” on page 41.

Performance Period

A performance goal, importance, and *duration* for a service class. You set up multiple performance periods for work that has changing performance requirements as work consumes more and more resources. You can specify up to eight performance periods.

Duration

Specifies the length of the period in service units. For a definition of service units, see Chapter 11, “Defining service coefficients and options,” on page 101. If the work included in this service class period does not complete when the number of service units have been used, the work moves into the next performance period. You do not specify a duration on the last defined period.

Response Time

The expected amount of time required to complete the work submitted under the service class, in milliseconds, seconds, minutes and hours. Specify either an average response time, or response time with a percentile. Percentile is the percentage of work in that period that should complete within the response time. Percentile must be a whole number. You must specify a system response time goal, not “end-to-end”. That is, workload management does not control all aspects of system performance, so response time scope is confined to the time workload management has control of the work. This time includes the time the work is using or waiting for CPU, storage, or I/O service.

Note: Workload management does not delay work, or limit it, to achieve the response time goal when extra processing capacity exists.

Velocity

A measure of how fast work should run when ready, without being delayed for WLM-managed resources. Velocity is a percentage from 1 to 99. See “Velocity formula” on page 52 for a description of the calculations needed to determine velocity.

Discretionary

Workload management defined goal. Associate this goal with work for which you do not have a specific performance goal. Work with a discretionary goal is run when excess resources are available.

Importance

The relative importance of the service class goal. Importance is a reflection of how important it is that the service class goal be achieved, Workload management uses importance only when work is not meeting its goal. Importance indicates the order in which work should receive resources when work is not achieving its goal. Importance is required for all goal types except discretionary. Importance applies on a performance period level and you can change importance from period to period. Importance is in five levels: 1 to 5, 1 being the highest importance.

CPU Protection

By specifying YES in the “CPU Critical” field when defining a service class,

you ensure that work of lower importance will always have a lower dispatch priority. See Chapter 12, “Defining special protection options for critical work,” on page 109 for more information.

I/O Priority Group

By specifying HIGH in the “I/O Priority Group” field when defining a service class, you ensure that work in this service class will always have a higher I/O priority than work in service classes assigned to I/O priority group NORMAL. See Chapter 12, “Defining special protection options for critical work,” on page 109 for more information.

Honor Priority

By specifying NO in the Honor Priority field, you explicitly prevent the overflow of specialty-engine-intensive work to standard processors. See Chapter 12, “Defining special protection options for critical work,” on page 109.

The values are:

DEFAULT

Current values of the IFAHONORPRIORITY and IIPHONORPRIORITY parameters in parmlib member IEAOPTxx are used when there is insufficient capacity on specialty engines (System z® Integrated Information Processors, or zIIPs, or System z Application Assist Processors, or zAAPs) for the workload demand in the service class. This is the default.

NO Independent of the current value of the IFAHONORPRIORITY and IIPHONORPRIORITY parameters in parmlib member IEAOPTxx, work in this service class is not allowed to overflow to standard processors when there is insufficient capacity on specialty engines (System z Integrated Information Processors, or zIIPs, or System z Application Assist Processors, or zAAPs) for the workload demand in the service class. The only exception is if it is necessary to resolve contention for resources with standard processor work.

The Honor Priority option is insignificant for discretionary service classes since work that is classified to these service classes never gets help from standard processors.

Velocity formula

The formula for velocity is:

$$\frac{\text{using samples}}{\text{using samples} + \text{delay samples}} \times 100$$

where:

using samples

include:

- The number of samples of work using the processor
- The number of samples of work using non-paging DASD I/O resources (in a state of device connect).¹

1. If I/O priority management is off, these samples are *not* included.

The I/O samples are derived from actual time measurements.

delay samples

include:

- The number of samples of work delayed for the processor
- The number of samples of work delayed for storage
- The number of samples of work delayed for non-paging DASD I/O.¹

Work delayed for storage includes:

- Paging
- Swapping
- Multiprogramming level (MPL)
- Server address space creation delays
- Initiation delays for batch jobs in WLM-managed job classes.

MPL is the SRM-controlled function that adjusts the number of address spaces allowed to be in central storage and ready to be dispatched.

I/O delays include:

- IOS queue
- Subchannel pending
- Control unit queue delays

The samples for subchannel pending and control unit queue delay are derived from actual time measurements.

Defining performance goals

This section explains how to define performance goals in your service definition. If you have an SLA today, you should consider a few things:

- Does it contain end-to-end response time?
If it does, then you need to keep in mind that workload management processes towards system response times, and make the adjustment when you define the performance goal. Section “Determining system response time goals” explains how you can determine the system response times of work.
- For what type of workloads do you need a different goal?
You may have some throughput type goals, which you need to convert into either response time goals, or velocity goals.

Determining system response time goals

Goal mode introduces several changes in the definition of a work request's response time. The changes more accurately reflect end-user expectations of response time.

The number of batch transactions equals number of jobs. Defining a response time goal may not be appropriate for some types of batch work, such as jobs with very long execution times. Work that is appropriate for a response time goal should have at least three transaction completions per 20 minutes of elapsed time. If there are too few completions, use a velocity or discretionary goal.

TYPRUN=HOLD and TYPRUN=JCLHOLD times are *not* included in batch response times.

Examples of service classes with response time goals

- A service class representing TSO/E work with multiple periods.

Service Class

TSO

Period 1

Response Time

85% 0.5 second

Importance

1

Duration

400 Service Units

Period 2

Response Time

80% 1 second

Importance

3

Duration

1000 Service Units

Period 3

Response Time

60% 15 second

Importance

4

Note that the percentile in period 1 and 2 refer to the transactions ending in each period, not the total TSO/E transactions.

- A service class representing CICS transactions.

Service Class

CICSHOT

Period 1

Response Time

0.5 second AVG

Importance

1

- A service class representing IMS transactions.

Service Class

IMSCAT1

Response Time

95% .3 Second

Importance

1

- A service class representing IMS transactions.

Service Class

OIMSCAT3

Response Time

5 sec AVG

Importance

3

Examples of service classes with response time goals

- A service class representing TSO/E work with multiple periods.

Service Class

TSO

Period 1

Response Time

85% 0.5 second

Importance

1

Duration

400 Service Units

Period 2

Response Time

80% 1 second

Importance

3

Duration

1000 Service Units

Period 3

Response Time

60% 15 second

Importance

4

Note that the percentile in period 1 and 2 refer to the transactions ending in each period, not the total TSO/E transactions.

- A service class representing CICS transactions.

Service Class

CICSHOT

Period 1

Response Time

0.5 second AVG

Importance

1

- A service class representing IMS transactions.

Service Class

IMSCAT1

Response Time

95% .3 Second

Importance

1

- A service class representing IMS transactions.

Service Class
OIMSCAT3

Response Time
5 sec AVG

Importance
3

Defining velocity goals

This section describes where to find information to set a velocity goal, and what kind of work is appropriate for velocity goals. Velocity goals define the acceptable amount of delay for work when work is ready to run. Velocity goals are intended for subsystems which use address spaces or enclaves to represent individual work requests. Velocity goals are not supported for work in the IMS and CICS subsystem work environments because velocity data is accounted to the region, not to the individual transaction. Velocity is a goal to consider for long-running jobs.

For a service class with multiple periods, you cannot switch from a velocity goal in one period to a response time goal in a later period. See “Subsystem support for goal types and multiple periods” on page 18 for a list of subsystems for which you can specify multiple periods.

Velocity goals are more sensitive to configuration changes than response time goals and should be monitored and adjusted when required after configuration changes. These configuration changes include:

- Change to the physical configuration, such as a new processor type.
- Changes to the capacity that is available to a system.
- Changes to the logical configuration, such as significant changes to the number of online processors, or implementation of HiperDispatch, or implementation of multi-threading

Adjusting velocity goals based on samples included in velocity calculation

You can adjust your velocity goals based on whether or not the following samples are to be included in the velocity calculation:

- I/O samples (included in the velocity calculation if I/O priority management is turned on)
- Initiation delay samples (included in the velocity calculation if you have WLM-managed batch initiators).

In the RMF Monitor I workload activity report, there are two fields, **I/O PRTY** and **INIT MGMT**, which indicate the following:

I/O PRTY

If you have I/O priority management turned off, then the **I/O PRTY** value shows you what your velocity would be if you were to turn I/O priority management on.

INIT MGMT

If you are not currently using WLM-managed batch initiators, then the **INIT MGMT** value shows you what your velocity would be if you turned over control of all batch initiators in this service class to WLM.

Note: For both the I/O PRTY and INIT MGMT fields, it is assumed that the *other* setting is unchanged. For example, the INIT MGMT field assumes that your current I/O priority management setting remains the same.

These fields may help you to adjust a current velocity goal in anticipation of including these samples.

Using velocity goals for started tasks

Velocity goals are the most appropriate goal for started tasks and long running work. Instead of figuring out a specific velocity goal for your started tasks, you should divide your started tasks into a high, a medium, and a low importance service class, and define a velocity that suffices for each category.

You can also take advantage of the system supplied service classes for started tasks: SYSTEM and SYSSTC. Workload manager recognizes special system address spaces (like GRS, SMF, CATALOG, MASTER, RASP, XCFCAS, SMXC, CONSOLE, IOSAS, WLM), puts them into the SYSTEM service class, and treats them accordingly. Address spaces in the SYSSTC service class are kept at a very high dispatching priority.

Note: You can also assign address spaces to the SYSTEM and SYSSTC service classes as part of your work classification rules. See “System-provided service classes” on page 88.

For information about how to define service classes and associated classification rules for started tasks, see “Using the system-supplied service classes” on page 95.

Velocity is also appropriate for the “server” started tasks, that is, the address spaces that do work on behalf of a transaction manager or resource manager, such as CICS AOR, or an IMS control region. Since the server address spaces are processing work that also has an assigned performance goal, the velocity goal that you assign to servers applies only during address space startup. Then workload management manages resources to meet the goals defined for the work the servers are processing, and not towards the goals defined for the servers.

If you have a version of a work manager such as CICS and IMS that does not support workload management, you cannot define a goal to the work manager's transactions, but you can define a velocity goal for its server address spaces.

Using discretionary goals

With discretionary goals, workload management decides how best to run this work. Since workload management's prime responsibility is matching resources to work, discretionary goals are used best for the work for which you do not have a specific performance goal. For a service class with multiple performance periods, you can specify discretionary only as the goal in the last performance period.

Discretionary work is run using any system resources not required to meet the goals of other work. If certain types of other work are overachieving their goals, that work may be “capped” so that the resources may be diverted to run discretionary work. See “Migration considerations for discretionary goal management” on page 166 for more information on the types of work that are eligible for resource donation, and how you may want to adjust those goals.

Examples of service classes with discretionary goals

- Discretionary goal as last period goal

Service Class
 DEVBATCH

Period 1

Response Time
 80% 1 minute

Importance
 2

Duration
 2000 Service Units

Period 2

Response Time
 80% 5 minutes

Importance
 3

Duration
 10000 Service Units

Period 3

Discretionary

- Discretionary goal for leftover work

Service Class
 ASDBATCH

Discretionary

Using performance periods

Performance periods are available for work that has variable resource requirements and for which your goals change as the work uses more resources. You specify a goal, an importance, and a duration for a performance period. Duration is the amount of service that period should consume before going on the next goal. Duration is specified in service units. For more information about defining durations, see Chapter 11, “Defining service coefficients and options,” on page 101.

You can define multiple performance periods for work in subsystems which use address spaces or enclaves to represent individual work requests. For a list of subsystems for which you can specify multiple periods, see “Subsystem support for goal types and multiple periods” on page 18.

Multiple periods are not supported for work in the IMS and CICS subsystem work environments because service units are accumulated to the address space, not the individual transactions. So, the system cannot track a duration for those transactions. Multiple periods are also not supported for work in the SYSH subsystem.

Defining goals appropriate for performance periods

As you go from one performance period to the next, you can change the type of goal. Goals should become less stringent going from one period to the next. A prime example would be changing to a velocity or discretionary type goal in the last period.

Using importance levels in performance periods

Importance levels should stay the same or decrease as the transactions move from one performance period to the next. Remember that importance applies only if a goal is not being met during the duration of the period.

Examples of multiple performance period goals

- Decreasing stringency of goal and decreasing importance from one period to the next

Service Class = BATCHX

Period 1

Velocity = 50

Importance = 3

Duration = 2500 SU

Period 2

Velocity = 15

Importance = 5

Chapter 9. Defining classification rules

Classification rules are the rules you define to categorize work into service classes, and optionally report classes, based on *work qualifiers*. A work qualifier is what identifies a work request to the system. The first qualifier is the subsystem type that receives the work request.

There is one set of classification rules in the service definition for a sysplex. They are the same regardless of what service policy is in effect; a policy cannot override classification rules. You should define classification rules after you have defined service classes, and ensure that every service class has a corresponding rule.

The full list of work qualifiers and their abbreviations is:

AI	Accounting information
CAI	Client accounting information
CI	Correlation information
CIP	Client IP address
CN	Collection name
CT	Connection type
CTN	Client transaction name
CUI	Client userid
CWN	Client workstation name
ESC	zEnterprise service class name from a Unified Resource Manager performance policy
LU	Logical Unit name
NET	Netid
PC	Process name
PF	Perform
PK	Package name
PN	Plan name
PR	Procedure name
PRI	Priority
PX	Sysplex name This is the same as cluster name for the SYSH subsystem.
SE	Scheduling environment name
SI	Subsystem instance
SPM	Subsystem parameter
SSC	Subsystem collection name
SY	System name
TC	Transaction class/job class
TN	Transaction name/job name

UI Userid

Note:

1. Not all work qualifiers are valid for every subsystem type; they are subsystem dependent. For details about which qualifiers are valid for which subsystems, see Table 7 on page 67.
2. For many of the qualifiers, you can specify *classification groups* by adding a G to the type abbreviation. For example, a transaction name group would be TNG. See "Using groups" on page 93 for more information.

A singular classification rule consists of a work qualifier and an associated service class or report class. You can also have multiple classification rules.

Example of a classification rule

```
Subsystem Type . : IMS          Fold qualifier names?  Y (Y or N)
Description . . . IMS medium interactive
-----Qualifier-----
Action  Type      Name      Start          Service      Report
-----
_____ 1 _____          _____          _____          _____
                                DEFAULTS: IMSMED
```

Note: The **Fold qualifier names** option, set to the default Y, means that the qualifier names will be folded to uppercase as soon as you type them and press Enter. If you set this option to N, the qualifier names will remain in the same case as they are typed. Leave this option set to Y unless you know that you need mixed-case qualifier names in your classification rules.

This example shows that all work coming into any IMS subsystem is associated with service class IMSMED. Service class IMSMED is the default service class for the IMS subsystem type. You can also assign a default report class to a subsystem type.

Since you might not want all work coming into a subsystem assigned to the same service class, or the same report class, you can specify multiple classification rules.

Figure 15 shows two classification rules. In the example, the incoming work request has work qualifiers of subsystem type, job name, job class, accounting information, and user ID.

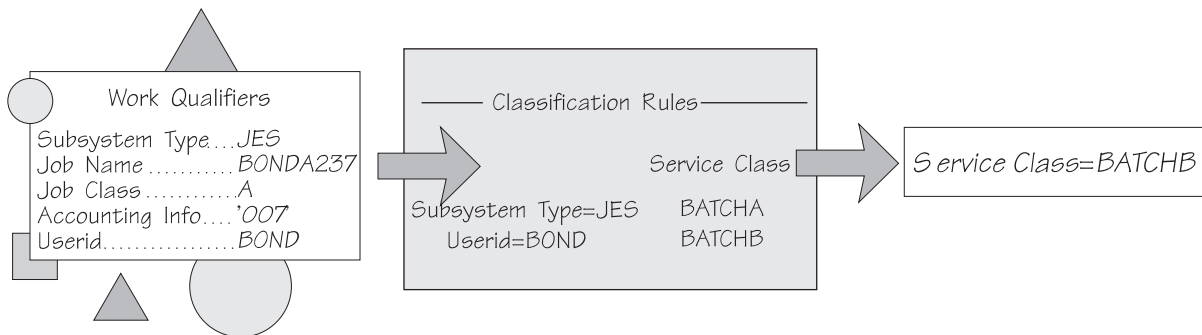


Figure 15. Using classification rules to assign work to service classes

In the example, the service administrator set up classification rules to assign all work coming into JES into service class BATCHA, unless the work has a user ID of BOND, in which case, it should be assigned to service class BATCHB. For JES classification, you do not need to specify JES2 or JES3.

Example of multiple classification rules

If you want all CICS work to go into service class CICSB except the following:

- You want work originating from LU name LONDON to run in service class CICS D
- You want work originating from LU name PARIS to run in service class CICS A, unless:
- The work is from the PAYROLL application, in which case you want it to go into service class CICS C.

You could specify the following classification rules:

Subsystem Type CICS

-----Qualifier-----		-----Class-----	
Type	Name	Service	Report
		DEFAULT: CICS B	_____
1	LU LONDON	CICS D	_____
1	LU PARIS	CICS A	_____
2	TN PAYROLL	CICS C	_____

This example has two classification rules with level 1 qualifiers: LU name LONDON and LU name PARIS. Under PARIS, there is a level 2 qualifier with transaction name PAYROLL. The PAYROLL qualifier applies only to transactions associated with the level 1 qualifier of PARIS.

In this example, if a work request comes in from an LU name other than LONDON or PARIS, it is assigned to the CICS B service class. If another work request comes in from Paris and is from the payroll application, it is assigned to the CICS C service class. If a work request is from the payroll application but came in from a system in London, then it is assigned to the CICS D service class.

The order of the nesting and the order of the level 1 qualifiers determine the hierarchy of the classification rules. The application supports eight characters for each rule. For more information about defining the hierarchy of the classification rules, see “Defining the order of classification rules” on page 85.

Defining classification rules for each subsystem

Work qualifiers depend on the subsystem that first receives the work request. When you are defining the rules, start with the service classes you have defined, and look at the type of work they represent. Determine which subsystem type or types process the work in each service class. Then understand which work qualifiers your installation could use for setting up the rules. It may be that your installation follows certain naming conventions for the qualifiers for accounting purposes. These naming conventions could help you to filter work into service classes. Also, understand which work qualifiers are available for each subsystem type. You can then decide which qualifiers you can use for each service class.

The following table shows the IBM-supplied subsystem types that workload management supports, the kind of work they run, whether they use address space-oriented transactions or enclaves (see special note for CICS and IMS), and

where to go for more information. (Unless otherwise noted, look for “Workload Manager” in each book.) A comparison of the various transaction types is shown in Table 6 on page 66.

Note: Enclaves are transactions that can span multiple dispatchable units in one or more address spaces. See *z/OS MVS Programming: Workload Management Services* for more information on enclaves.

Table 5. IBM-defined subsystem types

Subsystem type	Work description	Enclave, address space, or LPAR	For more information, see...
ASCH	The work requests include all APPC transaction programs scheduled by the IBM-supplied APPC/MVS transaction scheduler.	Address Space	<ul style="list-style-type: none"> • <i>z/OS MVS Planning: APPC/MVS Management</i>
CB	The work requests include all WebSphere Application Server client object method requests.	Enclave	<ul style="list-style-type: none"> • The online information included with the WebSphere Application Server system management user interface
CICS	The work requests include all transactions processed by CICS Version 4, and higher.	See note.	<ul style="list-style-type: none"> • <i>CICS Performance Guide</i> • <i>CICS Dynamic Transaction Routing in a CICSplex</i>
DB2	The work requests include only the queries that DB2 has created by splitting a single, larger query and distributed to remote systems in a sysplex. The local piece of a split query, and any other DB2 work, is classified according to the subsystem type of the originator of the request (for example, DDF, TSO, or JES).	Enclave	<ul style="list-style-type: none"> • <i>DB2 Data Sharing: Planning and Administration</i>
DDF	The work requests include all DB2 distributed data facility (DB2 Version 4 and higher) work requests.	Enclave	<ul style="list-style-type: none"> • <i>DB2 Data Sharing: Planning and Administration</i>
EWLM	Work requests include DB2 distributed data facility (DDF) requests that originate from an ensemble, through virtual servers that are classified within a Unified Resource Manager performance policy.	Enclave	<ul style="list-style-type: none"> • “Defining z/OS WLM classification rules for performance policies from the Unified Resource Manager” on page 79
IMS	The work requests include all messages processed by IMS Version 5 and higher.	See note.	<ul style="list-style-type: none"> • <i>IMS Administration Guide: System</i>
IWEB	The work requests include all requests from the world-wide-web being serviced by the Internet Connection Server (ICS), Domino Go Webserver, or IBM HTTP Server Powered by Domino (IHS powered by Domino). These requests also include those handled by the Secure Sockets Layer (SSL). This also includes transactions handled by the Fast Response Cache Accelerator.	Enclave	
JES	The work requests include all jobs that JES2 or JES3 initiates.	Address Space	<ul style="list-style-type: none"> • <i>z/OS JES2 Initialization and Tuning Guide</i> • <i>z/OS JES3 Initialization and Tuning Guide</i>
LDAP	The work requests include all work processed by the z/OS LDAP server	Enclave	<ul style="list-style-type: none"> • <i>z/OS IBM Tivoli Directory Server Administration and Use for z/OS</i>
LSFM	The work requests include all work from LAN Server for MVS.	Enclave	
MQ	The work requests include MQSeries Workflow work such as new client server requests, activity executions, activity responses, and subprocess requests.	Enclave	

Table 5. IBM-defined subsystem types (continued)

Subsystem type	Work description	Enclave, address space, or LPAR	For more information, see...
NETV	The work requests include NetView® network management subtasks and system automation (SA) subtasks created by Tivoli® NetView for z/OS.	Enclave	<ul style="list-style-type: none"> • <i>The Tivoli NetView for z/OS Tuning Guide</i> • <i>Tivoli NetView for z/OS Installation</i> • <i>Tivoli NetView for z/OS Administration Reference</i> • APAR OW54858
OMVS	The work requests include work processed in z/OS UNIX System Services forked children address spaces. (Work that comes from an enclave is managed to the goals of the originating subsystem.)	Address Space	<ul style="list-style-type: none"> • <i>z/OS UNIX System Services Planning</i>
SOM	The work requests include all SOM client object class binding requests.	Enclave	<ul style="list-style-type: none"> • <i>z/OS SOMobjects Configuration and Administration Guide</i>
STC	The work requests include all work initiated by the START and MOUNT commands. STC also includes system component address spaces such as the TRACE and PC/AUTH address spaces.	Address Space	<ul style="list-style-type: none"> • “Using the system-supplied service classes” on page 95
TCP	The work requests include work processed by the z/OS Communications Server.	Enclave	<ul style="list-style-type: none"> • <i>z/OS Communications Server: IP Configuration Guide</i>
TSO	The work requests include all commands issued from foreground TSO sessions.	Address Space	<ul style="list-style-type: none"> • <i>z/OS MVS Initialization and Tuning Guide</i> (Look for the discussion of TSO/E transactions in the “System Resources Manager” information.)
SYSH	Identifies non z/OS partitions (e.g. LINUX partition) in the LPAR cluster which needs to be managed by WLM according to business goals set for the partition.	LPAR	<ul style="list-style-type: none"> • “Non-z/OS partition CPU management” on page 27 and Chapter 18, “The Intelligent Resource Director,” on page 173

Important note about CICS and IMS transactions

CICS and IMS do not use enclaves, but use a different set of WLM services to provide transaction management.

CICS and IMS transactions can be assigned only response time goals (either percentile or average) within single period service classes. If you do not define any goals at all for CICS or IMS work, then the work will be managed to the velocity goals of the address spaces. Once you have defined a transaction goal for CICS or IMS work, then *all* subsequent work will be managed to those transaction goals, *not* to the velocity goals of the address spaces.

For example, you may initially be managing all CICS work to the velocity goals of the CICS address space. If you define a response time goal for a CICS transaction, you will be required to declare a default goal as part of that definition. Now all CICS transactions will be managed to those response time goals, even if they must accept the default.

Important note about NETV subsystem

Make sure to add the subsystem type NETV to your service definition (option 6 in ISPF application).

Tivoli NetView optionally allows you to let WLM manage NetView subtask performance in relation to other tasks and applications running on the system or sysplex. If enabled, NetView creates enclaves during subtask initialization and calls WLM to classify a subtask to the appropriate service class.

When a user decides to separate the management of NetView's network and system automation (SA) subtasks, NetView creates z/OS enclaves to manage those two sets of subtasks so that users can assign different performance goals to the enclaves. "Network" subtasks include all those not connected with system automation.

These two types of NetView enclaves should be classified to service classes with velocity goals. The goals should have approximately the same velocity value, but the goal assigned to NetView system automation enclaves should be more important than the goal assigned to any NetView network enclaves. There is no need to define a separate service class for NetView, if existing service classes in your service definition satisfy these conditions. For example, if SA z/OS or other system automation is used, a goal of Velocity = 50 and an Importance of 1 could be assigned. For non-system automation NetView subtasks, a goal of Velocity = 30 and an Importance of 2 could be assigned to give preference to the system automation NetView subtasks.

If the NetView WLM support is enabled, the absence of classification rules for subsystem type NETV will result in the NetView enclaves being classified to service class SYSOTHER.

Note that the WLM ISPF application does *not* validate the classification attributes used in the classification rules for subsystem type NETV.

If you have a subsystem *not* included in either of these tables, check its documentation for the kind of work requests supported by workload management and the applicable work qualifiers.

Table 6 summarizes the key differences among the service classes for enclave transactions, address space-oriented transactions, and IMS/CICS transactions.

Table 6. Enclave transactions, address space-oriented transactions, and CICS/IMS transactions

Transaction type	Allowable goal types	Allowable number of periods	RMF (or other monitor) reporting
Address space-oriented	<ul style="list-style-type: none"> Response Time Execution Velocity Discretionary 	Multiple	<ul style="list-style-type: none"> IOC, CPU, MSO, and SRB service consumption reported Execution delays reported Special reporting data provided. See note 1.
Enclave	<ul style="list-style-type: none"> Response Time Execution Velocity Discretionary 	Multiple	<ul style="list-style-type: none"> CPU service consumption reported Execution delays reported "Served by" reported for enclaves using TCBS Special reporting data provided. See note 1.
CICS/IMS	<ul style="list-style-type: none"> Response Time 	1	<ul style="list-style-type: none"> No service consumption reported (reported under regions) No execution delays reported (reported under regions) "Service Classes Being Served" reported (for service classes assigned to the server address spaces) "Response Time Breakdown in Percentage" reported Special reporting data provided. See note 1.

Note 1: See “Defining special reporting options for workload reporting” on page 82 for more information on special reporting.

The ISPF application provides these subsystem types as a selection list on the classification rules panel. You can add any additional subsystem type if it supports workload management on the same panel.

Defining work qualifiers

The name field for work qualifiers is 8 characters long. You can use nesting for the work qualifiers that run longer than 8 characters, which are the following:

- Accounting information
- Client accounting information
- Client IP address
- Client transaction name
- Client user ID
- Client workstation name
- Collection name
- Correlation information
- Package name
- Procedure name
- Process name
- Scheduling environment
- Subsystem parameter
- zEnterprise service class name

See “Organizing work for classification” on page 89 for how to nest by using the start position.

You can use masking and wildcard notation to group qualifiers that share a common substring. For work qualifiers that run longer than 8 characters, you can use a start position to indicate how far to index into the character string. When no start parameter is specified, WLM matches the name field for work qualifiers that run longer than 8 characters according to the number of characters specified. See “Organizing work for classification” on page 89 for details on how WLM matches the name field for work qualifiers.

Table 7 shows which work qualifiers are supported by each IBM-defined subsystem.

Table 7. Work qualifiers supported by each IBM-defined subsystem type

	A S C H	C B	C C S	D B 2	D D F	E W L M	I M S	I W E B	J E S	L D A P	L S F M	M Q	N E T V	O M V S	S O M	S T C	T C P	T S O	S Y S H
Accounting information	●			●	●			●						●		●		●	
Client accounting information				●	●														

Table 7. Work qualifiers supported by each IBM-defined subsystem type (continued)

	A S C H	C B	C I C S	D B 2	D D F	E W L M	I M S	I W E B	J E S	L D A P	L S F M	M Q	N E T V	O M V S	S O M	S T C	T C P	T S O	S Y S H
Client IP address				●	●														
Client Transaction Name				●	●		●												
Client user ID				●	●														
Client workstation name				●	●														
Collection name		●		●	●										●				
Connection type			●	●	●		●												
Correlation information				●	●														
LU name			●	●	●		●						●						
Netid				●	●		●												
Package name				●	●														
Perform				●					●							●		●	
Plan name				●	●														
Priority				●					●			●	●						
Procedure Name				●	●														
Process name				●	●							●							
Scheduling environment name				●					●										
Subsystem collection name				●	●				●										

Table 7. Work qualifiers supported by each IBM-defined subsystem type (continued)

	A S C H	C B	C I C S	D B 2	D D F	E W L M	I M S	I W E B	J E S	L D A P	L S F M	M Q	N E T V	O M V S	S O M	S T C	T C P	T S O	S Y S H
Subsystem instance		●	●	●	●		●	●	●	●	●	●					●		
Subsystem parameter				●	●			●				●			●	●			
Sysplex name	●	●	●	●	●		●	●	●	●	●	●		●	●	●		●	●
System name	●													●		●		●	●
Transaction Class / job class	●	●	●	●			●	●	●			●	●						
Transaction name / job name	●	●	●	●			●	●	●	●	●	●	●	●		●	●		
User ID	●	●	●	●	●		●	●	●			●	●	●	●	●		●	
zEnterprise service class						●													

Note: For subsystem type STC, you can use Subsystem Parameter as a work qualifier. However, you cannot use Subsystem Parameter Groups as a work qualifier.

For information about which qualifiers are used by other subsystems that are supporting workload management, see their supporting documentation.

Accounting information

ASCH The information passed on the JOB statement.

DB2 Accounting information is associated with the originator of the query; for example, the accounting information from TSO, JES, or DDF.

DDF Accounting information is the value of the DB2 accounting string associated with the DDF server thread.

JES The information passed on the JOB statement, not the EXEC statement.

Note: WLM/SRM ignores this if the JES2 init deck has a JOBDEF statement with parm ACCTFLD=OPTIONAL specified.

OMVS

Accounting data is normally inherited from the parent process of a z/OS UNIX System Services address space. In addition, when a daemon creates a process for another user, accounting data is taken from the WORKATTR of the RACF® user profile. A user can also assign accounting data by

setting the `_BPX_ACCT_DATA` environment variable or by passing accounting data on the interface to the `_spawn` service. For more information about z/OS UNIX System Services accounting information, see *z/OS UNIX System Services Planning*.

- STC** The information passed on the `JOB` statement.
- TSO** The accounting information specified by the TSO/E user when logging on to the system.

Because JCL supports 143 characters in accounting information, and the application allows only eight characters per rule, the application allows “nesting” for accounting information. See “Organizing work for classification” on page 89 for more information.

Example of nesting accounting information

In this example, you can “nest” accounting information (AI) in the classification rules.

```
Subsystem Type . . . . . JES (Required)
Description . . . . . All batch rules
```

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULTS: BATH00EY	_____
1	AI	43876AAA 1		_____
2	AI	DEPT58* 9	BATHBEST	_____

This example shows the classification rules for the JES subsystem. You can classify with more than the allowed 8 characters by nesting accounting information. In the example, all work with accounting information ‘43876AAADEPT58’ starting in position 1 for 14 characters is associated with service class BATHBEST.

Client accounting information

- DB2** The client accounting information associated with the originator of the query. Provided by the client information specified for the connection; may be different from the accounting information of TSO, JES, or DDF.
- DDF** The client accounting information associated with the DDF server thread. Provided by the client information specified for the connection; may be different from the DB2 accounting string associated with the DDF server thread.

Client IP address

- DB2** The source client IPv6 address associated with the originator of the query. Provided by the client information specified for the connection. The address must be left-justified and represented as a colon hexadecimal address. An example of an IPv6 address is '2001:0DB8:0000:0000:0008:0800:200C:417A'. The compressed format is not supported as for classifications. `SOURCELU`, `CLIENTIPADDR` and `NETID` are mutually exclusive.
- DDF** The source client IPv6 address associated with the DDF server thread. Provided by the client information specified for the connection. The address must be left-justified and represented as a colon hexadecimal address. An example of an IPv6 address is '2001:0DB8:0000:0000:0008:0800:200C:417A'. The compressed format is not supported as for classifications. `SOURCELU`, `CLIENTIPADDR` and `NETID` are mutually exclusive.

Client transaction name

- DB2** The client transaction name for the work request. Provided by the client information that is specified for the connection; might be different from the transaction or job name that is associated with the originator of the query.
- DDF** The client transaction name for the work request. Provided by the client information that is specified for the connection.
- IMS** The name of the Transaction Pipe (TPIPE).

Client user ID

- DB2** The client userid associated with the originator of the query. Provided by the client information specified for the connection; may be different from the user ID from TSO, JES or DDF.
- DDF** The client userid associated with the DDF server thread. Provided by the client information specified for the connection; may be different from the DDF server thread's primary AUTHID.

Client workstation name

- DB2** The client workstation name or host name associated with the originator of the query. Provided by the client information specified for the connection.
- DDF** The client workstation name or host name associated with the DDF server thread. Provided by the client information specified for the connection.

Collection name

- CB** The logical server group name defined using the WebSphere Application Server system management utility. This represents a set of WebSphere Application Server objects that are grouped together and run in a logical server. For more information, see the online information included with the WebSphere Application Server system management user interface.
- DB2** The collection name associated with the originator of the query; for example, the collection name from DDF.
- DDF** The DB2 collection name of the first SQL package accessed by the distributed relational database architecture (DRDA[®]) requestor in the work request.
- SOM** The logical server name defined using the SOM REGIMPL utility (defined to REGIMPL as the application alias). This represents a set of SOM objects that are grouped together and run in a logical server. For more information, see *z/OS SOMobjects Configuration and Administration Guide*.

Connection type

- CICS** The name of the TCP/IP service that received the request for this transaction.
- DB2** Connection type that is associated with the originator of the query; for example, the connection type from DDF.
- DDF** The DB2 connection type of the DDF server thread. The thread contains the value 'DIST' indicating it is a server.
- IMS** The port number of the TCP/IP service that received the request for this transaction.

Correlation information

- DB2** Correlation ID associated with the originator of the query; for example, the correlation ID from DDF.
- DDF** The DB2 correlation ID of the DDF server thread.

LU name and netid

LU name and netid are used mostly for qualifying CICS, DB2, DDF, and IMS work. If you want to filter on the fully qualified name, you can use the LU name value of up to 8 characters, and then define a sub-rule of netid for up to 8 characters.

- CICS** Only the LU name is available for CICS. It is the 8-byte NETNAME of the principal facility of the transaction instance. For details of the value of this parameter for non-VTAM terminals, and for transaction-routed transactions see *CICS/ESA Performance Guide*.
- DB2** The LU name and netid associated with the originator of the query; for example, the LU name and netid from DDF. For more information about the format of the LU name, see *DB2 SQL Reference*.
- DDF** The VTAM LU name and netid of the system that issued the structured query language (SQL) request. For more information about the format of the LU name, see *DB2 SQL Reference*.
- IMS** If a transaction comes in from an LU 6.2 device, you can specify both the LU name and the netid. Otherwise, you can specify only the 8 byte LU name.
- NETV** The LU name and the netid associated with the NetView subtask.

Package name

- DB2** Package name of the originator of the query; for example, the package name from DDF.
- DDF** The name of the first DB2 package accessed by the DRDA requestor in the work request.

Perform

- DB2** The performance group number for the thread associated with the query.
- JES** The performance group number specified using the PERFORM keyword on the JCL JOB statement.
- STC** One of the following:
- The performance group number specified using the PERFORM keyword on the START command.
 - The performance group number specified using the PERFORM keyword on the JCL JOB statement.
- TSO** The performance group number specified on the logon panel.

Plan name

- DB2** Plan name associated with the originator of the query; for example, the plan name from DDF.
- DDF** The DB2 plan name associated with the DB2 server thread. For DB2 private protocol requestors and DB2 Version 3 or higher DRDA requestors, this is the DB2 plan name of the requesting application. For non-DB2 requestors and other DRDA requestors, this is not applicable.

Priority

DB2 Priority associated with the originator of the query; for example, the priority from a batch job.

JES A value between 0 and 15, the priority associated with the batch job submitted through JES2 or JES3.

Note: For JES work, 15 is the highest priority and 0 is the lowest. (Contrast with MQ work, where 0 is the highest priority and 9 is the lowest.)

MQ A value between 0 and 9, the priority associated with the Websphere MQ Workflow message.

Note: For MQ work, 0 is the highest priority and 9 is the lowest. (Contrast with JES work, where 15 is the highest priority and 0 is the lowest.)

NETV Indicates a number between 1 and 9 that defines the dispatching priority of this task in relation to other subtasks running in this NetView program. For more information, see *Tivoli NetView Administration Reference*.

When you use priority as a work qualifier, you can use operators such as greater-than ('>') and less-than ('<') to group a range of priorities into one service or report class.

Priority Example

To put priority 8 and higher work into service class BATCH020, and put all other work into service class BATCH005, you would code the following:

```
Subsystem Type . . . . . JES (Required)
Description . . . . . Job Priority
```

```
-----Qualifier-----          -----Class-----
Type      Name      Start          Service      Report
1 PRI     >=8                DEFAULTS: BATCH005  _____
                                      BATCH020    _____
```

Procedure name

DB2 Procedure name associated with the originator of the query; for example, the DB2 stored procedure name from DDF.

DDF If the first SQL statement issued by the DDF client is a CALL statement, this field contains the unqualified name of the DB2 stored procedure. In all other cases, this field contains blanks.

Process name

DB2 A client application name

DDF A client application name

MQ The MQWIH_ServiceName from the message's work information header.

Scheduling environment name

JES The scheduling environment name assigned to the job.

DB2 Scheduling environment name associated with the originator of the query.

DDF The user ID for the client. This can be different from the authorization ID used to connect to DB2. This information is for identification only, and is not used for any authorization.

Subsystem collection name

- JES** The XCF group name.
- DB2** Subsystem collection name associated with the originator of the query.
- DDF** The subsystem collection name is the name assigned by the subsystem to related groups of its work. For example, DB2 data sharing group name.

Subsystem instance

You can use subsystem instance to isolate multiple instances of a subsystem. For example, use subsystem instance if you have a CICS production system as well as a CICS test system.

- CB** The WebSphere Application Server specific short name.
- CICS** The VTAM applid for the subsystem instance. For more information, see *CICS/ESA Dynamic Transaction Routing in a CICSplex*.
- DB2** The subsystem type associated with the originator of the query; for example:
- TSO for requests from TSO/E
 - JES for requests from batch jobs
 - DDF for requests from DDF
- DDF** The DB2 server's MVS subsystem name. For more information about the name, see *DB2 Administration Guide*.
- IMS** The IMS subsystem name, as defined on the IMSID positional parameter in the IMS DFSMPR procedure. It is a 1- to 4-character value that uniquely identifies the control region. The generation default is IMSA. For more information, see *IMS/ESA System Definition Reference*
- IWEB** The subsystem name from the application environment definition. (Note that this is identical to bits 0-7 of the Subsystem Parameter qualifier for IWEB).
- JES** The JES2 or JES3 subsystem name from the IEFSSNxx parmlib member.
- LDAP** The z/OS LDAP server's job name. Needed to distinguish between different LDAP servers.
- LSFM** The procname of address space in which LAN Server for MVS is running.
- MQ** The MQSeries Workflow subsystem name.
- NETV** The NetView WLM subsystem name as coded in CNMSTYLE. This is usually the NetView domain name. For more information, see *Tivoli NetView Administration Reference*.
- TCP** The TCP/IP jobname. For further information, refer to *z/OS Communication Server IP Configuration Guide*.

Subsystem parameter

If you have a vendor or home-grown subsystem type that has a qualifier other than the IBM supported ones, it could use the subsystem parameter. You should check your subsystem documentation to determine whether your subsystem supports the subsystem parameter, and in what parameter format.

Because the subsystem parameter is up to 256 characters long, you can nest to use more than the limit of eight characters. See “Organizing work for classification” on page 89 for further information on how to nest using the start position.

DB2 The subsystem parameter, if any, associated with the originator of the query.

DDF The subsystem parameter. This qualifier has a maximum length of 255 bytes. The first 16 bytes contain the client's user ID. The next 18 bytes contain the client's workstation name. The remaining 221 bytes are reserved.

Note the following:

- If the length of the client's user ID is less than 16 bytes, use blanks after the user ID to pad the length.
- If the length of the client's workstation is less than 18 bytes, use blanks after the workstation name to pad the length.

IWEB A 47-byte string formatted as follows:

1-8 Subsystem name
9 Blank
10-24 Source IP address
25 Blank
26-40 Target IP address
41 Blank
42-47 Target port

For more information, see *Internet Connection Server User's Guide*.

MQ The 32-byte *application_environment_name* from the APPLICID attribute of the process definition associated with the WLM-managed queue.

SOM A 246-byte string consisting of two 123-byte fields:

- Field 1 — class name
- Field 2 — method name

For more information, see *z/OS SOMobjects Configuration and Administration Guide*.

STC Indicates the system-provided service class name that will be assigned if a started task created with the high dispatching priority, privileged, or system task attribute is not assigned to a service class.

Values:

- SYSTEM — Started task was created with high dispatching priority attribute.
- SYSSTC — Started task is privileged or is a system task.
- (blank) — Started task was not created with the high dispatching attribute, is not privileged, and is not a system task.

Note: Subsystem Parameter Groups can be used as a work qualifier for subsystem types DB2, DDF, IWEB, MQ, SOM but **not** for subsystem type STC. See “Using groups” on page 93 for further information.

Sysplex name

For all subsystem types, use the sysplex name qualifier if you have a common service definition in multiple sysplexes, and need to assign different service classes or report classes based on the specific sysplex in which the work is running.

System name

The system name qualifier is supported for address spaces whose execution system is known at classification time. Note that JES is not eligible for this qualifier, as the system on which classification occurs may not be the system on which the job is run. Subsystem-defined transactions (CICS/IMS) and enclave-based transactions are not bound to an execution system at classification time, and are therefore not eligible either.

ASCH The name of the execution system.

OMVS

The name of the execution system.

STC The name of the execution system.

TSO The name of the execution system.

SYSH The name of the execution system.

Transaction class / job class

ASCH The job class that is used for work selection.

CB Name resulting from mapping the URI to a name. For more information, see *WebSphere Application Server for z/OS: Installation and Customization*.

CICS The name of the transaction class to which this transaction, or transid, belongs.

DB2 The job or transaction class that is associated with the originator of the query; for example, the job class from JES.

IMS The CLASS keyword on the PGMTYPE=parameter in the APPLCNT macro. For more information, see *IMS/ESA System Definition Reference*.

IWEB The arbitrary class name that is specified in the APPLENV directive in the Webserver's administrative file. Using the filtering function in the webserver, you can assign transactions to transaction classes based on the requested URL. The transaction classes can then be assigned unique service classes that use this Transaction Class qualifier. This is probably the most useful qualifier for IWEB work, because of its flexibility.

For the function Fast Response Cache Accelerator for high performance handling of cached static web pages, you must classify work as described in the following paragraph. Otherwise, it is assigned the default service class for IWEB work.

The transactions that are handled by the Cache Accelerator are all joined to a single, long-lived enclave. This enclave should be assigned a unique transaction class (as specified on the Webserver FRCAWLMParms directive). This transaction class should then be assigned to a service class with a single period and a velocity goal in the service policy under the IWEB subsystem type. Neither response time goals nor multiple periods are appropriate for this work, as WLM is not aware of the individual Cache Accelerator requests. (Because each individual transaction is so trivial, it would cost more resource to manage them than to simply process them.) In RMF reports, you see zero ended transactions for the Cache Accelerator service class (assuming you have no other work that is running in this service class), but you see some amount of accumulated service for this single enclave.

JES The job class that is used for work selection.

- MQ** A value of either ONLINE, meaning that the server is immediately available, or BACKLOG, meaning that the message was queued pending availability of a server.
- NETV** The NetView subtask type. Valid types are AOST, DST, HCT, MNT, NNT, OPT, OST, and PPT. For more information, see *Tivoli NetView Customization: Using Assembler*.

Transaction name / job name

- ASCH** The jobname in the JCL JOB statement in the APPC/MVS transaction program (TP) profile.
- CB** The method name, for example, GET, POST, or DELETE. For more information, see *WebSphere Application Server for z/OS: Installation and Customization*.
- CICS** A parameter on many CICS commands. It is often referred to as the CICS transaction identifier, or *tranid*. For more information, see *CICS/ESA Resource Definition Guide*.
- DB2** The transaction or job name associated with the originator of the query; for example, the job name from JES.
- IMS** The CODE= parameter on the IMS TRANSACT macro. For more information, see *IMS/ESA System Definition Reference*.
- IWEB** The method name, for example, GET, HEAD, POST, DELETE, or PUT.
- JES** The jobname of the JES managed job. For example, you may run a CICS region as a batch job in your installation. You would define it in classification rules as a transaction name in the JES subsystem type.
- LDAP** The z/OS LDAP server's enclave transaction name:
 - Any transaction name that is also defined in the configuration file of the directory server.
 - GENERAL for all LDAP work that is not assigned a user-defined exception class.

For further information, refer to *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

- LSFM** One of the following:
 - LSFMMMTX - multi-media transactions
 - LSFMFITX - file transactions
 - LSFMAMTX - administration transactions
 - LSFMCMTX - communication transactions

- MQ** The MQWIH_ServiceStep value from the message's work information header.
- NETV** The NetView subtask module name. For more information, see the NetView TASK statement description in the *Tivoli NetView Administration Reference*.

OMVS

The jobname for the z/OS UNIX System Services address space. By default, fork and spawn set jobname values to the user ID with a number (1-9) appended. However, daemons or users with appropriate privileges can set the _BPX_JOBNAME environment variable to change the jobname for forked or spawned children. In this way, servers and daemons in z/OS UNIX System Services address spaces can easily be assigned performance attributes different than other z/OS UNIX System Services address spaces.

- STC** One of the following:
- The name as specified on the JOBNAME= parameter of the START command
 - The name specified on the MOUNT command
 - The system address space name
 - The name on the JOB statement.

For example, you may run your IMS regions as started tasks in your installation. You would define these as transaction names in your STC subsystem type in classification rules. However, if IMS V5.1 is present, the STC rule for IMS is ignored and the rules for the IMS subsystem type are used when IMS becomes a server.

- TCP** The z/OS Communications Server transaction name associated with data traffic being processed by z/OS Communications Server on an independent enclave that has been active for a relatively long period of time. Currently the only transaction name supported by z/OS Communications Server is TCPENC01 for IPsec traffic. For further information, refer to *z/OS Communication Server IP Configuration Guide*.

User ID

- ASCH** The user ID of the user requesting the APPC/MVS service.
- CB** The user ID of the user requesting the WebSphere Application Server service.
- CICS** The user ID specified at LOGON time, which is the RACF (or other access control facility) defined resource. For more information about CICS user IDs, see *CICS/ESA CICS-RACF Security Guide*.
- DB2** The user ID associated with the originator of the query; for example, the user ID from TSO, JES, or DDF.
- DDF** The DDF server thread's primary AUTHID, after inbound name translation.
- IMS** The user ID specified at LOGON time, which is the RACF (or other access control facility) defined resource.
- IWEB** The user ID of the web server address space (not the original requestor's user ID). Note that because this user ID will generally be the same for all transactions, using this qualifier for IWEB work will have limited usefulness.
- JES** The user ID specified on the JOB statement on the RACF USER keyword.
- MQ** The first 8 bytes of the 12-byte message header field MQMD_USERIDENTIFIER.
- NETV** The NetView subtask ID. For NetView OST and NNT subtasks this is the NetView operator ID. For more information, see *Tivoli NetView User's Guide*. For AOST (automation subtasks) this is the NetView operator ID on the AUTOTASK command that started the AOST. For more information, see *Tivoli NetView Command Reference*. For NetView DST and OPT subtasks this is the taskname on the NetView TASK statement. For more information, see the *Tivoli NetView Administration Reference*. The MNT subtask ID is MNT. The PPT subtask ID is the NetView domain ID concatenated with the characters "PPT". The HCT subtask ID is the same as its LU name which is specified on the HARDCOPY statement. For more information, see *Tivoli NetView Administration Reference*.

OMVS

The RACF user ID associated with the address space. This user ID is either inherited from the parent process or assigned by a daemon process (for example, the rlogin or telnet daemon). For more information about z/OS UNIX System Services user IDs, see *z/OS UNIX System Services Planning*.

SOM The user ID of the user requesting the SOM service.

STC The user ID assigned to the started task by RACF (or other access control facility).

TSO The user ID specified at LOGON time, which is the RACF (or other access control facility) defined user profile.

zEnterprise service class name (ESC) from a Unified Resource Manager performance policy

EWLM

The subsystem type associated with work requests that originate from virtual servers in an ensemble workload. This subsystem type and work qualifier enable administrators to assign zEnterprise service classes to WLM service classes and report classes.

Defining z/OS WLM classification rules for performance policies from the Unified Resource Manager

For virtual servers that run z/OS, you can map service classes from a Unified Resource Manager performance policy to z/OS WLM service classes to achieve end-to-end goal-based performance management for multitier applications. Through classification rules for the IBM-defined EWLM subsystem type, z/OS WLM administrators can assign work requests that originate from virtual servers in an ensemble workload to service classes and report classes on z/OS.

The EWLM subsystem work requests include DB2 distributed data facility (DDF) requests that originate from an ensemble, through virtual servers (or hosts) that are classified within a Unified Resource Manager performance policy. Unified Resource Manager performance policies have service classes that specify either discretionary or velocity performance goals that the Unified Resource Manager uses to manage virtual servers in the ensemble workload. When you correlate these zEnterprise service classes to WLM service classes, you enable z/OS WLM to manage the work differently than it does for local z/OS work:

- z/OS WLM assigns the incoming work request to the appropriate WLM service class immediately, instead of going through the z/OS classification process.
- z/OS WLM can manage the work processed by a z/OS subsystem, such as DDF, according to different performance goals than those used for local z/OS work.
- z/OS WLM also can assign the work requests originating from virtual servers in the ensemble to different report classes than those used for local z/OS work.

The configuration tasks necessary for mapping zEnterprise service classes to WLM services classes are described in the following. Ensemble workload administrators, performance management administrators, and z/OS WLM administrators collaborate to complete these tasks.

For the ensemble, perform the following steps:

1. Create a workload and add the z/OS virtual servers to it. Also enable guest platform management provider (GPMP) support for each z/OS virtual server.

2. Define policies, service classes, and classification rules for the workload, and schedule activation of the appropriate policy. Note that z/OS WLM restricts the length of service class names to 32 characters, so you need to observe this restriction when you define the names of zEnterprise service classes that you want to map to WLM service classes.
3. Configure and activate a guest platform management provider on virtual servers that support the tiers of a multitier application. Also enable application response measurement (ARM) for the IBM middleware running on those virtual servers.

Depending on the configuration requirements of the multitier applications that your company uses, you might need to configure guest platform management providers and enable ARM for middleware not only on the virtual servers running z/OS, but also on z/VM guests and virtual servers on a POWER® blade.

For z/OS images running on virtual servers, perform the following steps:

1. For each Unified Resource Manager workload that contains virtual servers running z/OS:
 - Decide whether corresponding WLM service classes are required for the service classes within the Unified Resource Manager performance policy. If so, define the new WLM service classes that set only response time goals for Unified Resource Manager work requests. These response time goals apply to all DB2 distributed data facility (DDF) requests that originate from ensemble virtual servers that are classified within a performance policy from the Unified Resource Manager.
 - Optionally, define one WLM service class as the default service class for work associated with a Unified Resource Manager workload policy.
2. To map zEnterprise service classes to their corresponding WLM service classes, modify the classification rules for the EWLM subsystem type. Use only the ESC work qualifier in the classification rules.
3. Optionally, specify WLM report classes to monitor the performance of Unified Resource Manager work requests separately from local z/OS work.

To map specific zEnterprise service classes to WLM service classes, you need to know the names of the zEnterprise service classes and the performance goals that they set. For example, suppose that a Unified Resource Manager performance policy defines performance goals for Internet sales. Within that policy is a service class named WebSales, containing classification rules that identify ensemble virtual servers that send work requests to z/OS images. For those virtual servers, the WebSales service class sets the business importance as high and the velocity goal as fast.

On z/OS, suppose that the z/OS WLM administrator defines two new WLM service classes for the work requests that arrive from z/OS virtual servers in the ensemble: ZMGRDEFA is the default service class for most work and GPFASTHI is the corresponding service class for the WebSales service class. Through these two WLM service classes, the z/OS WLM administrator defines different response time goals for the work requests that arrive from virtual servers in the same ensemble as the z/OS virtual servers.

To map the new WLM service classes to the zEnterprise service classes, the z/OS WLM administrator modifies the classification rules for the EWLM subsystem type. Figure 16 on page 81 illustrates a sample WLM ISPF panel through which you map

zEnterprise service classes to the corresponding WLM service classes:

```

Subsystem-Type  Xref  Notes  Options  Help
-----
Command ==>  Modify Rules for the Subsystem Type  Row 3 to 10 of 16
SCROLL ==> PAGE

Subsystem Type . : EWLM          Fold qualifier names?  N (Y or N)
Description . . . Rules based on E2E classif. data

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row R=Repeat    IS=Insert Sub-rule
                                   More ==>

Action  -----Qualifier-----  -----Class-----
Type   Name      Start  Service  Report
-----
_____ 1 ESC      WebSales 1  ZMGRDEFA  ZMGRREPT
_____ 1 ESC      StoreSal 1  GPFSTHI
_____ 3 ESC      es      9  GPFSTHST

```

Figure 16. Sample z/OS WLM Classification Rules Panel for Mapping Unified Resource Manager performance policies

On the left side of the WLM Classification Rules Panel, note the values under the **Qualifier** heading:

- In the **Type** column, ESC is the required work qualifier that you must use for the EWLM subsystem type.
- In the **Name** column, the names of several zEnterprise service classes are displayed. WebSales is at the top of the list. The entry for StoreSales illustrates how to enter a zEnterprise service class name that is longer than eight characters.

On the right side of the panel, note the values under the **Class** heading:

- In the **Service** column, the names of WLM service classes are displayed. ZMGRDEFA is the default service class for z/OS WLM to use for work requests that are associated with a zEnterprise service class other than those listed on the left side of the panel. GPFSTHI is listed across from its corresponding zEnterprise service class, WebSales.
- In the **Report** column, the name of a default report class is displayed. This name identifies the WLM report class that you can use to monitor performance data for work requests associated with all zEnterprise service classes.

When you use the ESC work qualifier for classification rules, you must observe the following restrictions:

- The ESC work qualifier is valid only for the EWLM subsystem type.
- A subrule must use the same work qualifier as its parent classification rule: ESC
- A classification rule does not require a corresponding service class, and the EWLM subsystem type does not require a default service class. If processing of the classification rule does not result in the assignment of a valid WLM service class, the usual subsystem type specific WLM classification rules are applied to assign a WLM service class.
- The name of each specified zEnterprise service class cannot be longer than 32 characters.
- zEnterprise service class names are mixed case (fold qualifier name = N) and WLM uses case-sensitive names during the classification process.

- WLM uses classification rules starting from the top of the list as shown on the modify subsystem panel, so WLM uses the first correlation it finds for a zEnterprise service class.
- The selected WLM service class must have a response time goal; a velocity or discretionary goal is not supported.
- The selected WLM service class cannot have a multi-period goal.
- The selected WLM service class cannot belong to a resource group.

Defining special reporting options for workload reporting

To simplify mobile workload reporting, you can specify which work is mobile work that is eligible for mobile workload pricing. You do this with classification rules. When you create the rules, scroll right twice (PF11) on the panel and supply values for the Reporting Attribute option, as shown in the following example.

```

          Modify Rules for the Subsystem Type          Row 1 to 3 of 3
Command ==>> _____ Scroll ==>> CSR

Subsystem Type . : CICS          Fold qualifier names?  Y  (Y or N)
Description . . . CICS rules

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row R=Repeat    IS=Insert Sub-rule
                                   <=== More

Action  -----Qualifier-----  Storage  Reporting  Manage Region
        Type      Name      Start  Critical  Attribute  Using Goals Of

_____ 1 TC      BANKING  _____  NO      MOBILE      N/A
_____ 1 TC      HR      _____  NO      NONE        N/A
_____ 1 TN      ACCT    _____  NO      NONE        N/A

```

The Reporting Attribute option is supported for all IBM-supplied subsystem types. You can specify these values:

- NONE, for all your work. This is the default.
- MOBILE, for mobile work.
- CATEGORYA, for a first general purpose subset of work. This is provided for future use.
- CATEGORYB, for a second general purpose subset of work. This is provided for future use.

WLM then tracks processor consumption separately for each value of the reporting attribute, and reports consumption at the service and report class level in the IWMWRCAA answer area that is provided by the WLM Workload Activity Collection Service, IWMRCOLL. See the topic about using the workload reporting services in *z/OS MVS Programming: Workload Management Services* for details. At the system level, WLM reports consumption in the SRM Resource Control Table IRARCT.

The Reporting Attribute is independent from the assigned service and report class. It can be used to report on subsets of work that is running in a service or report class, and on subsets of work that is running on the whole system. This eliminates the need for introducing new dedicated service and report classes for mobile workload reporting, and leaves your existing reporting processes unaffected.

Classification with the Reporting Attribute option can be based on any work qualifier that is supported for the subsystem types. As for other classification rules, classification groups can be used to keep the classification rules simple to read, and efficient to check.

Example: mobile classification

Suppose that you have a classification rule for your banking transactions, and part of the transactions flows in from mobile devices. To differentiate those transactions from “normal” transactions, insert a subrule as shown in this example:

```

Command ==>      Modify Rules for the Subsystem Type      Row 1 to 3 of 3
                  _____      Scroll ==> CSR

Subsystem Type . : CICS      Fold qualifier names?  Y  (Y or N)
Description . . . CICS rules

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before      D=Delete row  R=Repeat      IS=Insert Sub-rule
                                           More ==>
               -----Qualifier-----
Action  Type      Name      Start
is_    1 TC      BANKING  ___
      1 TC      HR      ___
      1 TN      ACCT    ___
                                           -----Class-----
                                           Service      Report
                                           DEFAULTS: CICSDFT
                                           CICSFAST   CICS_RC1
                                           CICSLOW    CICS_RC2
                                           CICSLOW    CICS_RC3

```

If all mobile banking transactions flowed in through TCP/IP service TCP001, your subrule would look like the following:

```

Command ==>      Modify Rules for the Subsystem Type      Row 1 to 4 of 4
                  _____      Scroll ==> CSR

Subsystem Type . : CICS      Fold qualifier names?  Y  (Y or N)
Description . . . CICS rules

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before      D=Delete row  R=Repeat      IS=Insert Sub-rule
                                           More ==>
               -----Qualifier-----
Action  Type      Name      Start
      1 TC      BANKING  ___
      2 CT      TCP001  ___
      1 TC      HR      ___
      1 TN      ACCT    ___
                                           -----Class-----
                                           Service      Report
                                           DEFAULTS: CICSDFT
                                           CICSFAST   CICS_RC1
                                           CICSLOW    CICS_RC2
                                           CICSLOW    CICS_RC3

```

Do not add a service or report class. This means that the service and report class of the parent rule is used, and your existing reporting processes are not affected. Then, you can scroll right twice (PF11) to complete the Reporting Attribute option, as shown in this example:

```

Modify Rules for the Subsystem Type          Row 1 to 4 of 4
Command ==> _____ Scroll ==> CSR

```

```

Subsystem Type . : CICS          Fold qualifier names?  Y (Y or N)
Description . . . CICS rules

```

```

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
                B=Before     D=Delete row R=Repeat   IS=Insert Sub-rule
                <=== More

```

Action	Type	Qualifier Name	Start	Storage Critical	Reporting Attribute	Manage Region Using Goals Of
___	1 TC	BANKING	___	NO	NONE	N/A
___	2 CT	TCP001	___	NO	MOBILE	N/A
___	1 TC	HR	___	NO	NONE	N/A
___	1 TN	ACCT	___	NO	NONE	N/A

WLM then tracks the total and mobile processor consumption for the service and report class of the banking transactions, and also at the system level.

If you have mobile banking transactions that flow in through several TCP/IP services, you can create a classification group for them, as shown in this example:

```

Create a Group          Row 1 to 10 of 10
Command ==> _____

```

Enter or change the following information:

```

Qualifier type . . . . . : Connection Type
Group name . . . . . : TCPMOBIL (required)
Description . . . . . : TCP Service Names for mobile TX
Fold qualifier names? . . . . Y (Y or N)

```

Qualifier Name	Start	Description
TCP001	___	_____
TCP004	___	_____
TCP008	___	_____
TCP009	___	_____
TCP143	___	_____
TCP762	___	_____
_____	___	_____

Use the group that you created in your subrule, as shown in the following example:

```

Modify Rules for the Subsystem Type          Row 1 to 4 of 4
Command ==> _____ Scroll ==> CSR

```

```

Subsystem Type . : CICS          Fold qualifier names?  Y (Y or N)
Description . . C=Copy      M=Move      I=Insert rule
                B=Before     D=Delete row R=Repeat   IS=Insert Sub-rule
                . CICS rules

```

Action	Type	Qualifier Name	Start	Service	Report
___	1 TC	BANKING	___	CICSDFT	_____
___	2 CTG	TCPMOBIL	___	CICFAST	CICS_RC1
___	1 TC	HR	___	CICSSLOW	CICS_RC2
___	1 TN	ACCT	___	CICSSLOW	CICS_RC3

Defining the order of classification rules

When the subsystem receives a work request, the system searches the classification rules for a matching qualifier and its service class or report class. Because a piece of work can have more than one work qualifier associated with it, it may match more than one classification rule. Therefore, the order in which you specify the classification rules determines which service classes are assigned.

Prior to z/OS V1R3, with the IEAICSxx, the system used a set order for searching the work qualifiers. With workload management, there is no set order. You define it. Only the subsystem type must be the first level qualifier for classification rules. You determine the rest of search order by the order in which you specify the classification rules. You can use a different hierarchy for each subsystem.

Example of defining an order of rules

Suppose you are defining your JES classification rules. For all work requests coming into the JES2 subsystem instance, you want to assign work with user ID MNGRBIG and jobname PERFEVAL to service class BATSLOW. All other work with user ID beginning with MNGR should be assigned to service class BATHOT. So in this case, the hierarchy for the JES subsystem is:

1. Subsystem type, because it is always the first
2. Subsystem instance
3. User ID
4. Job name

Defining a subsystem service class default

A service class default is the service class that is assigned if no other classification rule matches for that subsystem type. If you want to assign any work in a subsystem type to a service class, then you must assign a default service class for that subsystem — except for STC. You are not required to assign a default service class for STC, even if you assign started tasks to different service classes.

Optionally, you can assign a default report class for the subsystem type. If you want to assign work running in a subsystem to report classes, then you do not have to assign a default service class for that subsystem.

Using inheritance in classification rules

Keep in mind that if you leave out the service class or report class on a classification rule, the work inherits the service class or report class from the higher level rule.

You can use this to your advantage by using the qualifier that applies to the most service classes in the subsystem as your first level qualifier. You can then list the exceptions in the subsequent levels.

Example of using inheritance

For example, for CICS, the available qualifiers include subsystem instance, user ID, transaction name, and LU name.

```
Subsystem Type . . . . . : CICS
Description . . . . . : CICS subsystem
                        -----Class-----
                        Service   Report
DEFAULTS: CICSB          _____
```

1	UI	ATMA	CICSA	ATMA
2	TN	CASH	_____	CASHA
2	TN	DEPOSIT	_____	DEPOSITA
3	LU	WALLST	_____	BIGDEP
1	UI	ATMC	CICSC	ATMC
2	TN	CASH	_____	CASHC
2	TN	DEPOSIT	_____	DEPOSITC
3	LU	WALLST	_____	BIGDEP

In this example, the installation set up their user IDs for their CICS work according to the ATM setup that they have. Since all of their interactive work is related to the ATMs, they chose user ID as their first level qualifier. Then, they wanted to separate out their cash transactions from their deposit transactions for reporting purposes, so they set up a report class for each.

The transactions do not have a service class explicitly assigned to them, so they inherit the service class from the rule one level before.

In addition, for the deposit transactions, they wanted to separate out those deposits coming from ATMs on Wall St location, because that area had been having some service troubles. So they defined a report class at a level 3 under the DEPOSIT transactions for each ATM user ID.

You cannot nest all qualifier types within themselves. For example, if you choose user ID as a first level, you cannot use user ID as a second level, or sub-rule qualifier. Nesting is allowed only for qualifiers longer than 8 characters and their associated groups. These are:

- Accounting information
- Client accounting information
- Client IP address
- Client transaction name
- Client userid
- Client workstation name
- Collection name
- Correlation information
- Package name
- Procedure name
- Process name
- Scheduling environment
- Subsystem parameter
- zEnterprise service class name

See "Organizing work for classification" on page 89 for how to nest using the start position.

Keep in mind that the system sequentially checks for a match against each level one rule. When it finds a match, it continues just in that family through the level two rules for the first match. Similarly, if a match is found at any given level, then its sub-rules are searched for further qualifier matches. The last matching rule is the one that determines the service class and report class to be assigned.

Example 1: Using the order of rules

Suppose you have defined the following classification rules for your IMS work:

```

-----Class-----
Service   Report
DEFAULTS: IMSB   _____
           IMSA   _____
           IMSC   ____  _____
1 Tran Name 5128
2 LU Name  BERMU

```

Suppose the following kind of work requests enter the system:

```

Transaction 1 Transaction 2 Transaction 3
Tran Name    5128          6666          5128
LU Name      BERMU        BERMU        CANCUN

```

- Transaction 1 is assigned service class IMSC, because the transaction 5128 is from the BERMU LU name.
- Transaction 2 is assigned service class IMSB, the subsystem default, because it is not transaction 5128, and therefore the system never checks any sub-rules.
- Transaction 3 is assigned service class IMSA, because it is associated with LU name CANCUN.
- If you specified the classification rules as:

```

1 LU Name  BERMU          IMSC   _____
2 Tran Name 5128        IMSA   ____  _____

```

Then all work from LU BERMU is assigned in service class IMSC, except work with Tran name 5128.

Example 2: Using the order of rules

Suppose you have defined the following rules for your IMS work:

```

-----Qualifier-----
Type      Name
1 SI      IMST
2 TC      15
1 TC      15
1 SI      IMSM
2 TC      15
-----Class-----
Service   Report
DEFAULTS: PRDIMSR _____
           TRNIMSR _____
           TRNIMSNR _____
           PRDIMSNR _____
           MDLIMSR _____
           MDLIMSNR _____

```

- If a work request in transaction class (TC) 15 enters the system from subsystem instance (SI) IMST, it is assigned service class TRNIMSNR.
- If a work request in transaction class (TC) 15 enters the system from subsystem instance (SI) IMSM, it is assigned service class PRDIMSNR and not MDLIMSNR as you might expect.

This is because the level 1 classification rule

```

1 TC      15          PRDIMSNR _____

```

comes first and the more explicit classification rule comes second:

```

1 SI      IMSM        MDLIMSR _____
2 TC      15          MDLIMSNR _____

```

The system stopped on the first level one match that it encountered. You can re-order the rules so that this does not occur. Put the most explicit rule first and the more general rule second as shown here:

```

-----Qualifier-----
Type      Name
1 SI      IMST
2 TC      15
1 SI      IMSM
2 TC      15
1 TC      15
-----Class-----
Service   Report
DEFAULTS: PRDIMSR _____
           TRNIMSR _____
           TRNIMSNR _____
           MDLIMSR _____
           MDLIMSNR _____
           PRDIMSNR _____

```

System-provided service classes

If some work comes into a system for which there is no associated service class defined in the classification rules, workload management assigns it to a default service class. There are several such default service classes:

SYSTEM

For all system address spaces designated 'high dispatching priority' (X'FF') address spaces. The high dispatching priority address spaces include: MASTER, TRACE, GRS, DUMPSRV, SMF, CATALOG, RASP, XCFAS, SMXC, CONSOLE, IOSAS, and others. For a list of the high dispatching priority address spaces in your installation, see the RMF Monitor II report and look for the x'FF' dispatching priority.

You do not need to set up service classes for these system address spaces. Workload management recognizes these as special system address spaces and treats them accordingly.

If for some reason you do want to control these address spaces, you can do the following:

- Define a service class for them
- Set up a classification rule in the STC subsystem type which assigns the address space to a service class other than the default STC service class.

Note: To make sure that the system runs smoothly, certain address spaces cannot be freely assigned to all service classes. The following address spaces are always classified into service class SYSTEM, independently of the user defined classification rules:

- *MASTER*
- INIT
- WLM
- XCFAS
- GRS
- CONSOLE
- IEFSCHAS
- IXGLOGR
- SMF
- CATALOG
- SMSPDSE
- SMSPDSE1

When you assign a service class other than SYSTEM to a started task eligible for the SYSTEM service class, it loses the high dispatching priority attribute and runs at the dispatching priority of the assigned service class period. The high dispatching priority attribute can be restored by one of the following methods:

- You can use the RESET command to change the started task's service class to SYSTEM.
- You can change the classification rules to explicitly classify the started task to SYSTEM and activate a policy.

You can also assign work to the SYSTEM service class as part of your work classification rules. You can *only* do this, however, for classification rules in the STC subsystem type, and only for address spaces that are designated as "high dispatching priority" address spaces.

For more information about using SYSTEM in classification rules for started tasks, see “Using the system-supplied service classes” on page 95.

SYSSTC

For all started tasks not otherwise associated with a service class. Workload management treats work in SYSSTC just below special system address spaces in terms of dispatching.

You can also assign work to the SYSSTC service class as part of your work classification rules. You can do this for classification rules in the following subsystem types:

- ASCH
- JES
- OMVS (z/OS UNIX System Services)
- STC
- TSO

Some address spaces normally created when running MVS are neither high dispatching priority, privileged, nor a system task, such as NETVIEW. These address spaces must be explicitly assigned to a service class such as SYSSTC.

For more information about using SYSSTC in classification rules for started tasks, see “Using the system-supplied service classes” on page 95.

SYSSTC1 - SYSSTC5

The service classes SYSSTC1, SYSSTC2, SYSSTC3, SYSSTC4, and SYSSTC5 are provided for future z/OS support. Service class SYSSTC and the SYSSTCx service classes are congruent as far as management of work is concerned. Work assigned to any of these service classes is managed identically to work assigned to any other. Currently, there is no technical reason to choose SYSSTCx as an alternative to SYSSTC. Many displays, for example, SDSF displays, provide the service class assigned to an address space. It is possible that one might assign SYSSTCx to convey some meaning that would be similar to a report class.

SYSOTHER

For all other work not associated with a service class. This is intended as a ‘catcher’ for all work whose subsystem type has no classification. It is assigned a discretionary goal.

Organizing work for classification

There are some ways you can organize your qualifiers for easier classification. You can use masking or wildcard notation as a way of grouping work to the same service class or report class. Or, you could set up a qualifier group for any qualifier except Priority and zEnterprise service class name.

If you have more than five rules at a given level for the same classification qualifier within a subsystem type, there may be performance implications. Qualifier groups are quicker to check than a single rule, so it may make sense to use them for performance sensitive subsystems like CICS and IMS.

You can use the start position for qualifiers longer than 8 characters. Those qualifiers are:

- Accounting information
- Client accounting information
- Client IP address

- Client transaction name
- Client userid
- Client workstation name
- Collection name
- Correlation information
- Package name
- Procedure name
- Process name
- Scheduling environment
- Subsystem parameter
- zEnterprise service class name

The following sections explain each kind of notation.

Using masking notation

You can use masking notation to replace a single character within a qualifier. This allows any character to match the position in the rule. Use a % in the position where the character would be. You can use multiple masks successively for multiple character replacement. If you specify a mask at the end of a character string, it could match on a null value OR a single character.

Examples of masking notation

For example, suppose you have a naming convention for all users in your IS services department that all user IDs start with DEPT58, followed by a letter A-F (depending on which division), ended by an I. Suppose also, you would like to bill your IS services department separately. You could use masking notation in setting up the classification rules as shown below.

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULTS: BATREG	
1	UI	DEPT58%I	_____	DEPT58

In the example, all work in this subsystem is associated with service class BATREG, and all work from the IS services department is associated with the report class DEPT58.

Using wildcard notation

You can also use *wildcard* notation for multiple character replacement in a character string. The wildcard character is an asterisk (*). You can use the wildcard character as the last position of a character string, or by itself. If a character string contains an asterisk in a position other than the last, it is treated as a character; for example, if you specify a character string of CI*S, the third character in a matching character string *must* have an asterisk as the third character. An asterisk by itself indicates a match for all characters.

Examples of wildcard notation

For example, suppose your installation has a naming convention for your CICS AORS and TORS. You can use the following wildcard notation in your CICS classification rules. Note that the subsystem instance of CI*S is not wildcard notation, a matching subsystem instance must be CI*S.

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULTS: CICSSTC2	_____
1	TN	TOR*	CICSSTC1_	
1	TN	AOR*	CICSSTC3_	
1	SI	CI*S	CICSTEST_	

Important Note

Be careful when putting specific definitions below wildcards, which might cause an unwanted early match. In the example, the rule for TOR11 is useless, because a TOR11 transaction will match the TOR* rule before it.

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULTS: CICSSTC2	_____
1	TN	TOR*	CICSSTC1_	
1	TN	TOR11	CICSSTC4_	

Using the start position

For work qualifiers longer than 8 characters, you can use a start position to indicate how far to index into the character string for a match. For example, you can assign all TSO/E users in a department to the same service class, assuming you follow a naming convention for accounting information for the department. Work qualifiers that run longer than 8 characters are:

- Accounting information
- Client accounting information
- Client IP address
- Client transaction name
- Client userid
- Client workstation name
- Collection name
- Correlation information
- Package name
- Procedure name
- Process name
- Scheduling environment
- Subsystem parameter
- zEnterprise service class name

The ISPF administrative application provides the **Start** field where you can specify the starting position for work qualifiers longer than 8 characters. The name field for a work qualifier is 8 characters long. If you are matching on a string that is fewer than 8 characters using a start position, you must use wildcard notation (asterisk) at the end of the string. Otherwise, the qualifier is padded with blanks to be 8 characters, and the blanks are used when making a match.

Example 1: Using the start position

Assume that you want to associate all JES2 work from department DIRS with the service class JESFAST. You assigned the default for JES2 work as service class JESMED. If all JES2 accounting information from department DIRS has the characters 'DIRS' starting in the eighth position, you enter a rule with qualifier DIRS* to match on just the 4 characters. If you want to filter out those jobs with

the 8 characters 'DIRS' starting in the eighth position, you need another rule with qualifier DIRS to assign those jobs to JESMED. The example shows the rules:

```
Subsystem Type . . . . . : JES
Description . . . . . All JES2 service classes
```

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULT: JESMED	_____
1 AI	DIRS	8	JESMED	_____
1 AI	DIRS*	8	JESFAST	_____

In this case, all jobs that have accounting information with the 8 characters 'DIRS' starting in the eighth position are assigned to JESMED. All other jobs that have the 4 characters 'DIRS' starting in the eighth position are assigned to JESFAST. All other work that is coming into JES is assigned to service class JESMED.

When no start parameter is specified, WLM matches the name field for work qualifiers that run longer than 8 characters according to the number of characters specified. This is different from work qualifiers that are 8 characters long. For those qualifiers, the qualifier name is always padded with blanks to be 8 characters. Work qualifiers that are 8 characters long are the following:

- Connection type
- LU name
- Net ID
- Perform
- Plan name
- Priority
- Sysplex name
- Subsystem instance
- Subsystem collection name
- System name
- Transaction class
- Transaction name
- Userid

Example 2: Blank padding for long and short work qualifiers

Assume that you want to associate all JES2 work from accounts with numbers that start with 0201, and all work from user IDs starting with DEPT58 with the service class JESFAST. The example shows the rules:

```
Subsystem Type . . . . . : JES
Description . . . . . All JES2 service classes
```

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
			DEFAULT: JESMED	_____
1 AI	0201		JESFAST	_____
1 UI	DEPT58*		JESFAST	_____

Without a start position, WLM matches work qualifiers longer than 8 characters according to the number of characters specified. In the example, when matching the accounting information for 0201, WLM matches it as a 4 character string. Therefore, a job with accounting information '020175,D123' would match.

Groups are allowed for all work qualifiers except for Priority and zEnterprise Service Class.

Group types are specified by adding G to the type abbreviation. For example, a transaction group name group is indicated as TNG.

Group types are usually valid for the same subsystem types as the underlying work qualifiers they group. For details on which qualifiers are valid for which subsystems, see Table 8. The only exception to this rule is the Subsystem Parameter Group which is **not** valid for subsystem type STC although the underlying Subsystem Parameter work qualifier is.

Qualifier groups of more than 5 members are quicker to check than single instances in the classification rules. So if you have, for example, a long list of CICS or IMS transaction names that you want to group in a service class or report class, consider setting up a group.

Example 1: Groups

If you want to assign a large number of CICS transactions to the same service class, you can create a transaction name group (TNG). You name the group, for example *CICSCONV*, and list all the transaction names you want included in the group.

```
Qualifier type . . . . . : Transaction Name
Group name . . . . . : CICSCONV (required)
Description . . . . . : CICS Conversational Group
Fold qualifier names? . . . . Y (Y or N)
```

Qualifier Name	Start	Description
CDBC	___	_____
CDBI	___	_____
CDBM	___	_____
CEBR	___	_____
CECI	___	_____
CECS	___	_____
CEDA	___	_____
CEDB	___	_____

```
Qualifier type . . . . . : Transaction Name
Group name . . . . . : CICSLONG (required)
Description . . . . . : CICS Long-Running Transaction
Fold qualifier names? . . . . Y (Y or N)
```

Qualifier Name	Start	Description
CDB0	___	_____
CSGX	___	_____
CSNC	___	_____
CSNE	___	_____
CSSX	___	_____

Then you use those group names in the classification rules, as shown in this panel:

```
Subsystem Type . . . . . : CICS
Description . . . . . : CICS transactions
```

-----Qualifier-----			-----Class-----	
Type	Name	Start	Service	Report
DEFAULTS:				
1	TNG	CICSCONV ___	CICSMED	_____
1	TNG	CICSLONG ___	CICSCONV	_____
			CICSLONG	_____

For work qualifiers running longer than 8 characters, you can use a start position for each group member to indicate how far to index into the character string for a match. Note that the start position needs not be the same for all group members. Furthermore, groups of such long work qualifiers can be nested.

Example 2: Groups of long work qualifiers

Assume you want to associate all JES2 work from a certain group of accounts with service class JESSLOW, except if it originates in a certain group of departments, in which case JESFAST should be used. The example shows the rules:

```
Qualifier type . . . . . : Accounting Information
Group name . . . . . : ACCOUNTS (required)
Description . . . . . : Accounts for JESSLOW
Fold qualifier names? . . . . Y (Y or N)
```

Qualifier Name	Start	Description
020175	---	_____
030275	---	_____
040375	---	_____
060575	---	_____
070675	---	_____
080775	---	_____
090875	---	_____

```
Qualifier type . . . . . : Accounting Information
Group name . . . . . : FASTDEPT (required)
Description . . . . . : Department for JESFAST
Fold qualifier names? . . . . Y (Y or N)
```

Qualifier Name	Start	Description
PURCHASE	8	_____
SALES	8	_____
SHIPPING	8	_____
ITDEP*	11	_____
HRDEP*	11	_____

Then you use those group names in the classification rules, as shown in this panel:

```
Subsystem Type . : JES Fold qualifier names? Y (Y or N)
Description . . . CICS transactions
-----Qualifier----- -----Class-----
Type      Name      Start      Service      Report
                DEFAULTS: JESMED _____
1 AIG      ACCOUNTS  ___      JESSLOW     _____
2 AIG      FASTDEPT  ___      JESFAST     _____
```

A job with accounting information '040375,SHIPPING' is then assigned service class JESFAST. Similarly, a job with accounting information '070675,D71ITDEP' is assigned service class JESFAST. A job that contains the job statement '050475,CONTROL ' is assigned service class JESSLOW, because the department is not contained in the FASTDEPT group. A job with accounting information '020177,SALES ' is assigned the default service class JESMED, because the account number does not match group ACCOUNTS, and therefore the system never checks any sub-rules.

Using the system-supplied service classes

You can also take advantage of the system-supplied service classes to simplify the process of defining service classes and classification rules for started tasks.

Use the system-provided service classes SYSTEM, and SYSSTC for your STC service classes. WLM recognizes certain system address spaces when they are created (like GRS, SMF, CATALOG, MASTER, RASP, XCFAS, SMXC, CONSOLE,

IOSAS, WLM), puts them into the SYSTEM service class, and treats them accordingly. If a started task is not assigned to a service class, WLM manages the started task in the SYSSTC service class. Started tasks in SYSSTC are assigned a high dispatching priority. This is appropriate for started tasks such as JES and VTAM. Not all started tasks are appropriate for SYSSTC, as a CPU-intensive started task could use a large amount of processor cycles. However, if your processor is lightly loaded, or in a 6-way, 8-way, or 10-way MP, SYSSTC might be appropriate, because that one task may not affect the ability of the remaining processors to manage the important work with goals.

Note: *MASTER*, INIT and WLM always run in the SYSTEM service class and cannot be reassigned via the service definition.

The following example implements started task classification with these assumptions:

- Any started tasks not explicitly classified are given low priority.
- Started tasks are defined in three transaction name groups: HI_STC, MED_STC, and LOW_STC.
- System defaults are used for MVS-owned address spaces.
- Separate reporting is used for Master, DUMPSRV, and GRS.

```

Subsystem Type . . . . . : STC
Description . . . . . All started tasks
-----Qualifier-----
Type      Name      Start
1 TN      %MASTER%  ___
1 TN      GRS       ___
1 TN      DUMPSRV   ___
1 TNG     DB2       ___
.....
1 SPM     SYSTEM    ___
1 SPM     SYSSTC    ___
1 TNG     HI_STC    ___
1 TNG     MED_STC   ___
1 TNG     LOW_STC   ___
.....

```

DEFAULTS: DISC _____
 SYSTEM _____
 SYSTEM _____
 SYSTEM _____
 VEL60 _____

The first rule assigns *MASTER* to report class MASTER. (Note that "SYSTEM" is shown in the service class column. In this case, it is not a true assignment as *MASTER* must *always* run in the SYSTEM service class anyway.) *MASTER* could also be classified using the SPM SYSTEM rule, but since separate reporting is desired for *MASTER*, it is classified separately. GRS and DUMPSRV are handled similarly.

The SPM rules assign started tasks created with the high dispatching priority attribute to SYSTEM, and other privileged or system tasks to SYSSTC. This allows you to let MVS manage started tasks that it recognizes as special.

Notes:

1. Note that explicitly defining the SPM rules, as in the example at hand, is optional. If they were removed from the example, then the high dispatching priority work would still be assigned to SYSTEM, and the other privileged or system tasks would still be assigned to SYSSTC, as those are the defaults. The reason for explicitly defining them here with the SPM rules is to protect yourself from inadvertently assigning them elsewhere in the rules that follow the SPM rules.

- Note, also, that the placement of these SPM rules is crucial. If they had been declared first, then the three TN rules intended to assign *MASTER*, DUMPSRV, and GRS to report classes would have never been reached.

System tasks are those given the privileged and/or system task attribute in the IBM-supplied program properties table or in the SCHEDxx parmlib member. See the SCHEDxx chapter in *z/OS MVS Initialization and Tuning Reference* for a list of system tasks. An example for such a system task is DB's subsystem parameter module, the program DSNYASCP that is defined in the IBM-supplied PPT (IEFSDPPT) with the system task attribute. Without the fourth classification rule, the SPM rules would enforce all DB2 subsystem address spaces that execute this program (MSTR, DBM1, DIST, SPAS) to be classified into SYSSTC.

A transaction name group is defined to match all DB2 subsystem regions with 4-character subsystem names not already classified. Such a group for DB2 V4 could be defined as follows:

```
Qualifier type . . . . . : Transaction Name
Group name . . . . . : DB2      (required)
Description . . . . . : All non-IRLM DB2 regions
Fold qualifier names? . . . . Y (Y or N)
```

Qualifier Name	Start	Description
%%%DBM1	___	DB2 database services AS_____
%%%MSTR	___	DB2 master AS_____
%%%DIST	___	DB2 distributed data facility AS_____
%%%SPAS	___	DB2 stored procedure AS_____

Note: The **Fold qualifier names** option, set to the default Y, means that the qualifier names will be folded to uppercase as soon as you type them and press Enter. If you set this option to N, the qualifier names will remain in the case they are typed in. Leave this option set to Y unless you know that you need mixed case qualifier names in your classification rules.

Started tasks in the HI_STC transaction name group are run in service class SYSSTC, MED_STC in VEL35I3, and LOW_STC in VEL15I5. The goals on the service classes use importance to guarantee that started tasks of low value will be sacrificed if necessary to keep medium-value started tasks meeting goals. Note that this does not guarantee any relationship between the dispatching priorities that will be observed for service classes VEL35I3 and VEL15I5.

Started tasks which do not match any of the classification rules are assigned the default service class DISC. If the default service class were left blank, these started tasks would be assigned to SYSSTC.

Chapter 10. Defining report classes

Optionally, classification rules can assign incoming work to a report class. Workload management provides data for reporting on all of the service definition terms on a service class period and workload basis. Report classes can be used to report on a subset of transactions running in a single service class but also to combine transactions running in different service classes within one report class. In the first case, a report class is called homogeneous, in the second case it is called heterogeneous. Heterogeneous report classes can cause incorrect performance data, since the data collected is based on different goals, importance, or duration. WLM allows a caller of the workload reporting service IWMRCOLL to detect whether a report class is homogeneous or heterogeneous for a given report interval. See *z/OS MVS Programming: Workload Management Services* for more information on how a reporting product can receive an indication whether a report class is homogeneous or heterogeneous.

The data available for report classes include:

- Number of transactions completed
- Average response times
- Resource usage data
- State samples
- Response time distribution buckets
- Work manager delay data

Each item in the list is related to a report class period. The reporting products are advised to report response time distributions only for homogenous report class periods, where only 1 service class contributed data to the report class.

For CICS and IMS workloads, resource usage data and state samples are reported with the service classes for the regions, not the service classes assigned to the transactions.

Note that a report class can have as many periods as the largest service class within the same service definition has periods. Not all of them may be used, though. Report classes inherit periods automatically based on the transaction's service class period that is contributing to the report class. A reporting product can determine whether a report class is homogeneous within a given reporting interval and is advised to report the data as one period if the report class is heterogeneous. Based on its capabilities and options, a reporting product can report the data of a homogeneous report class as one period or as multiple periods.

For subsystem STC initiator INITs should never be in a report class.

You can assign up to a maximum of 2047 report classes, with no more than one report class per work request or transaction.

Name Report class name

Description

Description of the report class

Name (required)

Eight character identifier of the report class.

Description (optional)

An area of 32 characters to describe the report class.

Example 1: Defining report classes

Suppose you have defined the following rules for your CICS work:

```

Subsystem Type . . . . . : CICS
Description . . . . . : CICS subsystem

```

			-----Class-----	
			Service	Report
			DEFAULTS: CICSB	
1	UI	ATMA	CICSA	ATMA
2	TN	CASH		CASHA
2	TN	DEPOSIT		DEPOSITA
3	LU	WALLST		BIGDEP
1	UI	ATMC	CICSC	ATMC
2	TN	CASH		CASHC
2	TN	DEPOSIT		DEPOSITC
3	LU	WALLST		BIGDEP

In this example, the cash transactions are separated from their deposit transactions for reporting purposes. Report class ATMA therefore, does not include all transactions with userid ATMA, because it does not include the cash or deposit transactions.

Chapter 11. Defining service coefficients and options

Firstly, this information explains how *service* can be calculated in WLM (see “Calculating the amount of service consumed”).

Next, this information describes the options that must be specified for workload management. They are the following:

- How the system calculates the amount of resources that work consumes with service coefficients (see “Service definition coefficients” on page 102).
- Whether workload management is to dynamically set I/O priorities based on performance goals (see “Specifying I/O priority management” on page 104).
- Whether workload management is to consider I/O priority groups when dynamically setting I/O priorities (see “Enabling I/O priority groups” on page 104).
- Whether workload management is to dynamically reassign parallel access volume alias addresses based on performance goals (see “Specifying dynamic alias management” on page 105).

All of these options can be set using the Service Coefficient/Service Definition Options panel in the WLM ISPF application, as shown in “Working with service coefficients and options” on page 220.

Calculating the amount of service consumed

One of the basic functions of WLM/SRM is to monitor the dynamic performance characteristics of all address spaces under its control to ensure distribution of system resources as intended by the installation.

A fundamental aspect of these performance characteristics is the rate at which an address space is receiving *service* relative to other address spaces competing for resources within the same domain.

The amount of service consumed by an address space is computed by the following formula:

$$\begin{aligned} \text{service} = & (\text{CPU} \times \text{CPU service units}) \\ & + (\text{SRB} \times \text{SRB service units}) \\ & + (\text{IOC} \times \text{I/O service units}) \\ & + (\text{MSO} \times \text{storage service units}) \end{aligned}$$

Figure 17. Formula for Calculating Service Consumption

where CPU, IOC, MSO, and SRB are installation defined service definition coefficients and:

CPU service units =

task (TCB) execution time, multiplied by an SRM constant which is CPU model dependent. Included in the execution time is the time used by the address space while executing in cross-memory mode (that is, during either secondary addressing mode or a cross-memory call). This execution time is not counted for the address space that is the target of the cross-memory reference.

SRB service units =

service request block (SRB) execution time for both local and global SRBs, multiplied by an SRM constant which is CPU model dependent. Included in the execution time is the time used by the address space while executing in cross-memory mode (that is, during either secondary addressing mode or a cross-memory call). This execution time is not counted for the address space that is the target of the cross-memory reference.

I/O service units =

measurement of data set I/O activity and JES spool reads and writes for all data sets associated with the address space. SRM calculates I/O service using I/O block (EXCP) counts. When an address space executes in cross-memory mode (that is, during either secondary addressing mode or a cross-memory call), the EXCP counts or the DCTI will be included in the I/O service total. This I/O service is not counted for the address space that is the target of the cross-memory reference.

Storage service units =

(central page frames) × (CPU service units) × 1/50, where 1/50 is a scaling factor designed to bring the storage service component in line with the CPU component. NOT included in the storage service unit calculation are the central storage page frames used by an address space while referencing the private virtual storage of another address space through a cross service (that is, through secondary addressing or a cross-memory call). These frames are counted for the address space whose virtual storage is being referenced.

Service definition coefficients

The amount of system resources an address space or enclave consumes is measured in *service units*. Service units are calculated based on the CPU, SRB, I/O, and storage (MSO) service an address space consumes.

Service units are the basis for period switching within a service class that has multiple periods. The *duration* of a service class period is specified in terms of service units. When an address space or enclave running in the service class period has consumed the amount of service specified by the duration, workload management moves it to the next period. The work is managed to the goal and importance of the new period.

Because not all kinds of services are equal in every installation, you can assign additional weight to one kind of service over another. This weight is called a *service coefficient*.

Changing your coefficient values

Prior to z/OS V1R3, you could use the same coefficients as used in the IEAIP5xx parmlib member. Then you could directly compare RMF data, and determine your durations properly.

However, if you plan to use workload management, it is probably a good time for you to rethink your coefficients. The current defaults are inflated, given the size and processing capability of processors. Processors can consume much higher amounts of service, and as a result, service unit consumption numbers are very high. These high numbers can cause problems if they reach the point where they wrap in the SMF fields. If they wrap, you may see abnormally large transaction counts, and last period work may be restarted in the first period.

It is possible for you to make them smaller, yet still maintain the same relationship between the coefficient values. Consider changing your definitions to the following:

```
CPU  1
IOC  0.5
MSO  0
SRB  1
```

If you do decide to change the coefficients, you must re-calculate your durations and accounting procedures.

Tip: If you want to gather storage service information by service class, but don't want it affecting your durations or accounting procedures, use an MSO coefficient of 0.0001. This results in very low MSO service unit numbers, but still allows you to obtain storage service information through RMF.

Since changing the coefficients affects durations and accounting values, the defaults are meant to be consistent with settings seen in the field today. If you do not define the service coefficients, the defaults are:

```
CPU  10.0
IOC  5.0
MSO  0.0
SRB  10.0
```

Using the storage (MSO) coefficient for calculations

The MSO service definition coefficient is externalized in the following SMF records:

SMF type 30

Performance Section, Field SMF30MSC

SMF type 72 subtype 3

Workload Manager Control Section, Field R723MMSO

The externalized value is the value specified in the WLM administrative application, scaled up by 10 000. For example, if you specify MSO = 0.0001 the externalized value is 1; if you specify MSO = 1.0 the externalized value is 10,000.

The idea of storage service units is to account for central storage being held while CPU cycles are being used. The basic unit of measure is a page frame (4 096 bytes) held for one CPU service unit. To make MSO roughly commensurate with CPU service units, the raw number is divided by 50 to yield MSO service units.

By scaling the MSO value with 4096 and dividing it by 50, the internal value used by SRM becomes slightly less precise than the input value which is externalized in the SMF records. The following formula is used for scaling:

$$\frac{\text{Input MSO Coefficient scaled by 10000}}{1000} \times \frac{4096}{50} + 1$$

Figure 18. MSO Coefficient Formula

The result is truncated to the nearest integer value. Therefore, input values between 0.0001 and 0.0122 all result in a value of 1. An MSO coefficient of 1 results in a value of 82 which is used by SRM to calculate MSO service.

To use the MSO service definition coefficient for your own calculations, apply the same scaling in order to understand the storage service units shown in the SMF 30 and SMF Type 72 records.

Specifying I/O priority management

I/O priority queueing is used to control non-paging DASD I/O requests that are queued because the device is busy. You can optionally have the system manage I/O priorities in the sysplex based on service class goals.

The default for I/O priority management is **no**, which sets I/O priorities equal to dispatching priorities. If you specify **yes**, workload management sets I/O priorities in the sysplex based on goals.

WLM dynamically adjusts the I/O priority based on how well each service class is meeting its goals and whether the device can contribute to meeting the goal. The system does not micro-manage the I/O priorities, and changes a service class period's I/O priority infrequently.

When I/O priority management is on, I/O samples are used in the velocity formula. See "Velocity formula" on page 52 for more information.

Considerations for I/O priority management

If you specify I/O priority management, workload management dynamically sets I/O priorities based on goals and I/O activity, and includes the I/O information when calculating execution velocity. So you might see some changes in your velocity values. The recommended setting for I/O priority management is YES.

The new DASD I/O using and DASD I/O delay samples are reported even when I/O priority management is turned off. This allows you to calculate the new velocity values to plan for velocity changes.

Enabling I/O priority groups

I/O priority groups can be used to protect work which is extremely I/O-sensitive. When you assign a service class to I/O priority group HIGH, you ensure that work managed by this service class always has a higher I/O priority than work managed by service classes assigned to I/O priority group NORMAL which is the default for service classes.

The default for Enabling I/O Priority Groups is **no**, which will ignore I/O priority groups. If you specify **yes**, you enable workload management to consider I/O priority groups and setting higher I/O priorities for work in I/O priority group HIGH than for work in NORMAL.

Considerations for I/O priority groups

I/O priorities are also recognized at the I/O controller level. To ensure that all connected systems utilize the same ranges of I/O priorities consistently it is recommended that you enable I/O priority groups in the service definitions of all active systems and sysplexes sharing I/O controllers as soon as you exploit I/O

priority groups in one sysplex. In other words, you may have to enable I/O priority groups even if no service class is assigned to I/O priority group HIGH in the service definition.

Enabling I/O priority groups requires to specify **yes** for I/O priority management.

Specifying dynamic alias management

This section discusses the WLM and HCD considerations for dynamic alias management.

Workload management considerations for dynamic alias management

As part of the Enterprise Storage Subsystem's implementation of parallel access volumes, the concept of base addresses versus alias addresses is introduced. While the base address is the actual unit address of a given volume, there can be many alias addresses assigned to a base address, and any or all of those alias addresses can be reassigned to a different base address. With dynamic alias management, WLM can automatically perform those alias address reassignments to help work meet its goals and to minimize IOS queueing.

Note that to be able to move aliases to other bases, the aliases first have to have been initially assigned to bases via the ESS specialist, and the bases have to have come online at some point.

When you specify **yes** for this value on the Service Coefficient/Service Definition Options panel, you enable dynamic alias management globally throughout the sysplex. WLM will keep track of the devices used by different workloads and broadcast this information to other systems in the sysplex. If WLM determines that a workload is not meeting its goal due to IOS queue time, then WLM attempts to find alias devices that can be moved to help that workload achieve its goal. Even if all work *is* meeting its goals, WLM will attempt to move aliases to the busiest devices to minimize overall queueing.

In addition to automatically managing aliases, WLM ensures during system initialization that a minimum number of aliases is assigned to parallel access volumes with page data sets. The minimum number of aliases for a volume is:

$$2 * \text{number_of_page_datasets} - 1$$

This allows each page data set to have two I/Os active at a time, and ensures that paging intensive activities such as system dumping are not delayed by IOS queueing. This automatic enforcement of the minimum aliases only happens if dynamic parallel access volumes management is active for the device.

IMPORTANT: If you enable dynamic alias management, you must also enable I/O priority management. So you need to specify **yes** for both of these options on the panel. If I/O priority management is set to **no**, you will get only the efficiency part of dynamic alias management and not the goal-oriented management. This means that WLM will make alias moves that minimize overall IOS queueing, but these moves will not take service class goals into consideration.

HCD considerations for dynamic alias management

While you can globally enable or disable dynamic alias management on the WLM ISPF panel, you can also individually enable or disable dynamic alias management on a given device via HCD. You can do this by specifying WLMPAV=YES or NO in that device's HCD definition.

Note, however, that there is no consistency checking for dynamic alias management between different systems in a sysplex. If at least one system in the sysplex specifies WLMPAV=YES for a device, then dynamic alias tuning will be enabled on that device for *all* systems in the sysplex, even if other systems have specified WLMPAV=NO. It is recommended *not* to use dynamic alias management for a device unless all systems sharing that device have dynamic alias management enabled. Otherwise, WLM will be attempting to manage alias assignments without taking into account the activity from the non-participating systems.

Note, also, that you can specify WLMPAV=YES or NO on both base and alias devices. The WLMPAV settings on an alias device, however, is only meaningful when the alias device is bound to a base device that is offline, as follows:

- If the base device is *offline*, then only alias devices with WLMPAV set to YES will be reassigned to other base devices.
The WLMPAV setting on the base device itself is irrelevant when the base device is offline, for either "giving" or "receiving" aliases. (Even if WLMPAV was set to YES on the base device, it cannot have new aliases assigned to it, as it is offline.)
- For any base device that is offline to one or more systems in the sysplex and online to others, the WLMPAV keyword in HCD needs to be set to NO for the base device and its aliases. You need to statically assign the desired number of aliases for the base device via the ESS Specialist. If you try to use dynamic alias management for such a device, WLM will make unpredictable alias moves.
- If the base device is *online*, then the WLMPAV settings on the aliases are ignored, as follows:
 - If WLMPAV is set to YES on the base device, then the aliases can be reassigned regardless of their WLMPAV settings.
 - If WLMPAV is set to NO on the base device, then no aliases can be reassigned, regardless of their WLMPAV settings.

For a WLMPAV=YES base device, the aliases initially assigned to it should be allowed to default to YES. The only situation where you might want to change an alias to WLMPAV=NO is if the alias is initially assigned to a WLMPAV=NO base device. Because the base is set to NO, the aliases initially assigned to it will not be moved to other bases by WLM. Then, because the aliases are set to NO, if the base is ever varied offline, the aliases remain assigned to that base and cannot be reassigned by WLM to other bases. Certain combinations of WLMPAV settings are not recommended, as described in table Table 8:

Table 8. Effects of WLMPAV settings on base and alias devices

Base device WLMPAV setting	Alias device WLMPAV setting	Effects / recommendations
YES	YES	<ul style="list-style-type: none"> • If base is online: Base is WLM-managed. Aliases can be freely moved to and from the base device by WLM. • If base is offline: Aliases become unbound and are available to WLM to assign to other WLM-managed bases.

Table 8. Effects of WLMPAV settings on base and alias devices (continued)

Base device WLMPAV setting	Alias device WLMPAV setting	Effects / recommendations
YES	NO	Not recommended. If base is WLM-managed, then it is not predictable which aliases will remain bound to that base when the base goes offline. If the base device is set to YES, then you should set the aliases to YES as well. (See previous option.)
NO	YES	Not recommended. If the base is not WLM-managed, then you risk losing all of its aliases when the device goes offline. (See next option.)
NO	NO	<ul style="list-style-type: none"> • If base is online: Base is not WLM-managed. The initial aliases assigned to this base remain there. • If base is offline: Aliases remain bound to the offline base device and are not available to WLM for reassignment. When the base comes back online, it retains its initial alias assignments.

In order for dynamic alias management to be most effective, you should try to spread out your aliases in the initial definition. If one base device has several alias devices while other base devices have none, it will take more time for WLM to reassign the aliases appropriately. Ideally, you should have at least two aliases assigned to each base at the outset.

For more information about HCD definitions, see *z/OS HCD Planning*.

Chapter 12. Defining special protection options for critical work

Several options are available to help performance administrators protect critical work. Although applicable to several other subsystem types, CICS and IMS work will particularly benefit from the enhancements described in this section:

- Long-term storage protection
- Long-term CPU protection
- Long-term I/O protection
- Modifications of transaction response time management

These three options are described, and then illustrated by an example in “Sample scenarios” on page 114.

Important: The use of these options limits WLM's ability to manage the system. This may affect system performance and/or reduce the system's overall throughput.

Long-term storage protection

When you assign long-term storage protection to critical work, WLM restricts storage donations to other work. This option can be useful for work that needs to retain storage during long periods of inactivity because it cannot afford paging delays when it becomes active again. With long-term storage protection assigned, this work loses storage only to other work of equal or greater importance that needs the storage to meet performance goals.

You assign long-term storage protection with the “Storage Critical” option, found by scrolling right on the Modify Rules for the Subsystem Type panel:

```
====> _____ Modify Rules for the Subsystem Type _____ Row 1 to 2 of Command
====> _____ SCROLL ====> PAG

Subsystem Type . . : CICS          Fold qualifier names?  Y  (Y or N)
Description . . . : CICS Transactions

Action codes:  A=After  C=Copy  M=Move  I=Insert rule
                B=Before  D=Delete row  R=Repeat  IS=Insert Sub-rule
                <==== More

-----Qualifier-----
Action  Type  Name  Start  Storage  Reporting  Manage Region
Critical Attribute Using Goals Of
-----
_____ 1 TN   COMBL*  _____  NO   NONE   N/A
_____ 2 UI   COMBLD  _____  NO   NONE   N/A
_____ 2 UI   COMFTP  _____  YES  NONE   N/A
```

Figure 19. Specifying the **Storage Critical** Option

Storage critical for address spaces

You can assign storage protection to all types of address spaces using classification rules for subsystem types ASCH, JES, OMVS, STC, and TSO. By specifying YES in the **Storage Critical** field for a classification rule, you assign storage protection to

all address spaces that match that classification rule. An address space must be in a service class that meets these requirements, however, before it can be storage-protected:

- The service class must have a single period.
- The service class must have either a velocity goal, or a response time goal of over 20 seconds.
- The service class must not be connected to a resource group that has a memory limit defined.

Notes:

1. These requirements only apply to the address spaces classified under subsystem types ASCH, JES, OMVS, STC, and TSO.
2. While an address space which has the storage critical attribute joins an enclave, it loses the storage critical attribute.

Storage critical for CICS and IMS transactions

For CICS and IMS work, you can assign long-term storage protection by specifying YES in the **Storage Critical** field in the rules for specific transactions. Once you specify YES for one transaction in a CICS/IMS service class, *all CICS/IMS transactions in that service class* will be storage-protected. If a CICS or IMS region is managed as a server by WLM (managed to the response time goals of the transactions it serves) and *any* of the transaction service classes it serves is assigned storage protection, then the CICS/IMS region itself is automatically storage-protected by WLM.

As an alternative to assigning storage protection based on specific transaction service classes, you can instead choose to assign storage protection to the region in which the transactions run. You do this by adding or modifying the STC or JES classification rule that assigns the service class to the region.

Long-term CPU protection

When you assign long-term CPU protection to critical work, you ensure that less important work will generally have a lower dispatch priority. (There are some rare exceptions, such as when other work is promoted because it is holding an enqueue for which there is contention.) This protection can be valuable for work which is extremely CPU-sensitive, such as certain CICS and IMS transactions.

Use the **Cpu Critical** option on the Create or Modify a Service Class panel to assign long-term CPU protection to a specific service class:

Create a Service Class Row 1 to 2 of 2

Service Class Name	APPC9	(Required)
Description		
Workload Name	APPC	(name or ?)
Base Resource Group		
Cpu Critical	YES	(YES or NO)
I/O Priority Group	NORMAL	(NORMAL or HIGH)
Honor Priority	DEFAULT	(DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period, E=Edit period, D=Delete period.

Figure 20. Specifying the **CPU Critical** option

You can assign CPU protection to service classes handling address space-oriented work, enclave work, or CICS/IMS transactions, but the service class must have only one period, and it cannot have a discretionary goal. If a CICS or IMS region is managed as a server by WLM (managed to the response time goals of the transactions it serves) and any of the transaction service classes it serves is assigned CPU protection, then the CICS/IMS region itself is automatically CPU-protected by WLM

Long-term I/O protection

When you assign a service class to I/O priority group HIGH, you ensure that work managed by this service class always has a higher I/O priority than work managed by service classes assigned to I/O priority group NORMAL. This protection can be valuable for work which is extremely I/O-sensitive.

Use the **I/O Priority Group** field on the Create or Modify a Service Class panel and specify HIGH to assign long-term I/O protection to a specific service class.

Create a Service Class		Row 1 to 1 of 1
Service Class Name	CICSHI	(Required)
Description	_____	
Workload Name	CICSWKLD	(name or ?)
Base Resource Group	_____ (name or ?)	
Cpu Critical	NO	(YES or NO)
I/O Priority Group	HIGH	(NORMAL or HIGH)
Honor Priority	DEFAULT	(DEFAULT or NO)
Specify BASE GOAL information. Action Codes: I=Insert new period, E=Edit period, D=Delete period.		

Figure 21. Specifying the I/O Priority Group option

I/O priority group HIGH is ignored by workload management unless I/O priority groups are enabled (see “Enabling I/O priority groups” on page 104).

Honor priority

With parameters IFAHONORPRIORITY and IIPHONORPRIORITY in parmlib member IEAOPTxx, you control at the system level whether specialty engines get help from standard processors when there is insufficient capacity for the workload demand. The recommended default setting for the parameters is YES. While this makes sense for important workloads that offload work to specialty engines, like CICS or DB2, there might be individual workloads for which you want to prevent the overflow from specialty engines to standard processors, so that the overflow cannot negatively impact the work running on standard processors.

In the **Honor Priority** field on the Create or Modify a Service Class panel, specify NO to prevent overflow from specialty engines to standard processors for work in this service class.

```

                                Create a Service Class                                Row 1 to 1 of 1
Service Class Name . . . . . OFFONLY (Required)
Description . . . . . _____
Workload Name . . . . . CB (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO (YES or NO)
I/O Priority Group . . . . . NORMAL (NORMAL or HIGH)
Honor Priority . . . . . NO (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

```

Figure 22. Specifying the **High Priority** option

With this setting, specialty engine work that is running in the service class does not get help from standard processors, except if it is necessary to resolve contention for resources with standard processor work.

For transaction servers and enclave servers, consider the following:

- For CICS and IMS work, you can specify Honor Priority=NO for the service classes of the individual transactions running in a CICS or IMS region. While in a region that is not managed as a transaction server all work is always associated to a single service class, CICS and IMS regions can execute transactions that are classified into different transaction service classes. The region is then managed by WLM to the mix of service classes that it serves. In that case, only if you specify Honor Priority=NO for all transaction service classes that are running in the same region, work in that region is prevented from overflowing to standard processors. If any of the transaction service classes is assigned Honor Priority=DEFAULT, work in that region might overflow to standard processors.
- For enclave servers like WebSphere Application Server, which execute work within enclaves, the meaning of the Honor Priority option is the following:
 - Work that is running within an enclave is handled according to the definition of the enclave’s service class.
 - For work that is not running as part of an enclave, that is, work that is running in a TCB/SRB that has not joined an enclave, the meaning of the Honor Priority option depends on the setting of parameter MANAGENONENCLAVEWORK in parmlib member IEAOPTxx:
 - With MANAGENONENCLAVEWORK=NO, which is the default, work running outside of an enclave in the server address space is conceptually unmanaged. If you specify Honor Priority=NO for all service classes of the enclaves that are running in the same address space, non-enclave work is prevented from overflowing to standard processors.
 - With MANAGENONENCLAVEWORK=YES, non-enclave work in that address space is managed towards the goal of the service class of the address space. If the service class of the address space is defined with Honor Priority=NO, the non-enclave work in the address space is prevented from overflowing to standard processors.

Server address spaces can switch dynamically into and out of the “server” status, depending on what work they must do. If you want them to always keep the same value for the Honor Priority option, be sure that you also specify the option for the region service classes.

Modifications of transaction response time management

Use the **Manage Region Using Goals Of** field in the Modify Rules for the Subsystem Type panel to declare that a specific CICS/IMS region is not managed to the response times of the CICS/IMS transactions that it processes. Other regions are not affected by what is in this column, and that this option can be used only in STC and JES classification rules:

```

====> _____ Modify Rules for the Subsystem Type _____ Row 1 to 2 of Command
                SCROLL ==> PAG

Subsystem Type . : STC          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type
Action codes:  A=After  C=Copy  M=Move  I=Insert rule
                B=Before D=Delete row R=Repeat  IS=Insert Sub-rule
                <=== More

-----Qualifier-----
Action  Type      Name      Start      Storage  Reporting  Manage Region
-----  ---      ---      ---      ---      ---      ---
_____ 1 SY       SYST1     _____ NO        NONE      TRANSACTION
_____ 2 TN       CICSTEST  _____ NO        NONE      REGION
_____ 2 TN       CICS*    _____ YES       NONE      TRANSACTION
_____ 2 TN       TOR      _____ NO        NONE      BOTH
_____ 2 TN       AOR*    _____ NO        NONE      TRANSACTION
  
```

Figure 23. Specifying the **Manage Region Using Goals Of** option

If you specify TRANSACTION in this field (the default), the region is managed as a CICS/IMS transaction server by WLM. If you specify REGION in this field, the region is managed to the performance goal of the service class that is assigned to that region (address space). In other words, it is *not* managed as a CICS/IMS transaction server by WLM.

If you specify BOTH in this field, the region is also managed to the performance goal of the service class that is assigned to that region, but it nevertheless tracks all transaction completions so that WLM can still manage the CICS service classes with response time goals. Option BOTH should be used only for CICS TORs. All AORs should remain at the default TRANSACTION.

If you specify TRANSACTION or BOTH, RMF reports performance information as follows:

- Response time data is reported in the WLMGL SCPER report for the service class in which those transactions are running
- Response time data is also reported in the WLMGL RCLASS report for the report class in which those transactions are running
- The service classes that are served by the region are reported in the WLMGL SCLASS report for the service class in which the region is running.

If you specify REGION, only the information in the SCPER report changes. Transaction response times reported by these regions is not reported in any service class. This response time data is still reported in the RCLASS report for the transaction report class, and the service classes that are served by the region are still reported in the SCLASS report for the service class in which the region is running. In both of these cases, this information is useful if you are migrating the CICS/IMS region to transaction response time management, and need both a transaction response time benchmark and a list of the service classes the region is serving.

Sample scenarios

The following scenarios illustrate different configurations and how they would benefit from these options.

Many of the panels shown in this section are actually composites of the information that is displayed after scrolling right from the Modify Classification Rules panels.

Scenario 1

In this scenario, you want to assign storage protection and/or CPU protection to address spaces.

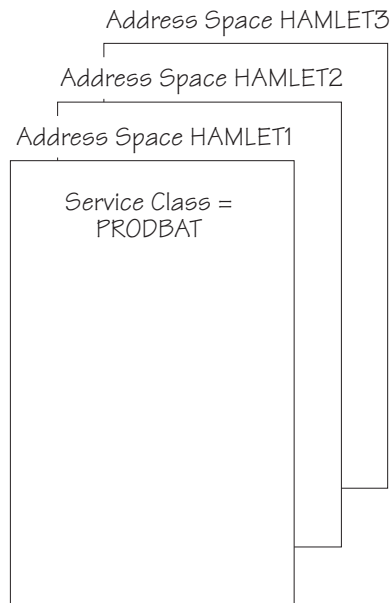


Figure 24. Scenario 1: Address Spaces

Suppose that you have the following classification rules:

Subsystem Type . . :		JES		Fold qualifier names? Y (Y or N)			
Description . . .		IBM-defined subsystem type					
Action	Type	-----Qualifier-----	-----Class-----	Storage			
___	1	TN	HAMLET1	___	PRODBAT	___	NO
___	1	TN	HAMLET*	___	PRODBAT	___	NO

To assign storage protection to the address space HAMLET1, change the **Storage Critical** value to field to YES in the classification rule for HAMLET1:

Subsystem Type . . :		JES		Fold qualifier names? Y (Y or N)			
Description . . .		IBM-defined subsystem type					
Action	Type	-----Qualifier-----	-----Class-----	Storage			
___	1	TN	HAMLET1	___	PRODBAT	___	YES
___	1	TN	HAMLET*	___	PRODBAT	___	NO

To assign storage protection to the address spaces HAMLET2 and HAMLET3 (but not HAMLET1), change the **Storage Critical** value to field to YES in the wildcard rule for HAMLET*:

```

Subsystem Type . : JES          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

Action  Type  Name  Start  Service  Report  Storage
-----  ---  ---  ---  ---  ---  ---
   1  TN  HAMLET1  ___  PRODBAT  ___  NO
   1  TN  HAMLET*  ___  PRODBAT  ___  YES
  
```

In this instance, HAMLET1 is not protected, as it matches the HAMLET1 rule first. To protect all of the address spaces, you would specify YES in both rules.

The default values (with no CPU protection assigned) in the service class definitions would be:

```

Service Class Name . . . . . PRODBAT (Required)
Description . . . . . Production Batch_____
Workload Name . . . . . JES (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO (YES or NO)
  
```

To assign CPU protection to all of these address spaces, change the value in the **Cpu Critical** field in the service class definition for PRODBAT:

```

Service Class Name . . . . . PRODBAT (Required)
Description . . . . . Production Batch_____
Workload Name . . . . . JES (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . YES (YES or NO)
  
```

Scenario 2

The next few example scenarios will use the CICS/IMS regions shown in Figure 25.

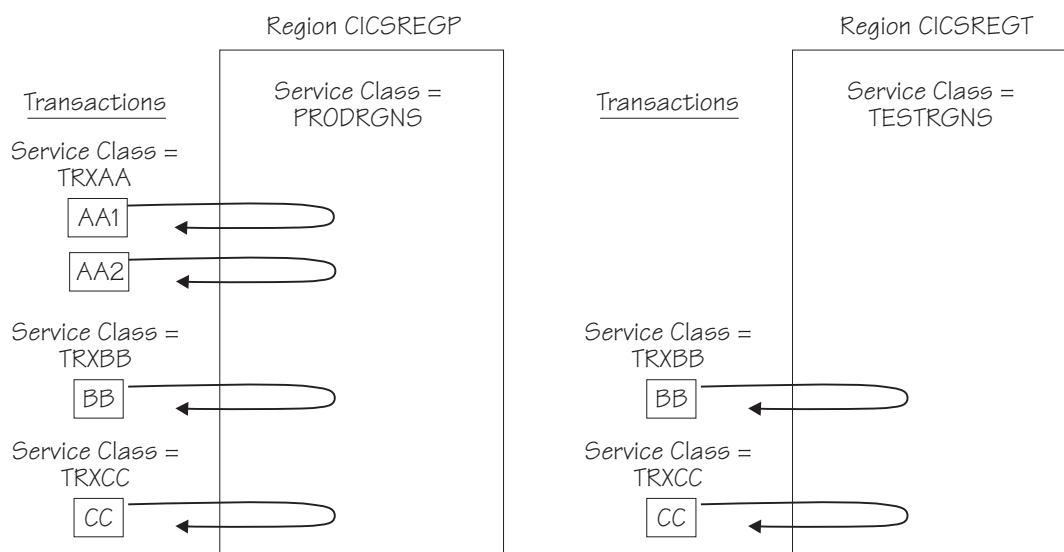


Figure 25. Scenarios 2, 3, 4, 5: CICS/IMS regions

Suppose you have the following CICS classification rules:

```

Subsystem Type . . : CICS          Fold qualifier names?  Y  (Y or N)
Description . . . IBM-defined subsystem type

```

Action	-----Qualifier-----			-----Class-----		Storage
	Type	Name	Start	Service	Report	Critical
___ 1	TN	AA1	___	TRXAA	AA1RPT	NO
___ 1	TN	AA2	___	TRXAA	AA2RPT	NO
___ 1	TN	BB	___	TRXBB	___	NO
___ 1	TN	CC	___	TRXCC	___	NO

Suppose you have the following STC classification rules to classify the CICS regions:

```

Subsystem Type . . : STC          Fold qualifier names?  Y  (Y or N)
Description . . . IBM-defined subsystem type

```

Action	-----Qualifier-----			-----Class-----		Storage	Manage Region
	Type	Name	Start	Service	Report	Critical	Using Goals Of:
___ 1	TN	CICSREGP	___	PRODRGNS	___	NO	TRANSACTION
___ 1	TN	CICSREGT	___	TESTRGNS	___	NO	TRANSACTION

Suppose the transaction service classes are defined as follows. (Only TRXAA is shown here. The definitions for TRXBB and TRXCC would also have NO specified in the **Cpu Critical** field.)

```

Service Class Name . . . . . TRXAA  (Required)
Description . . . . . Transactions AA1, AA2 _____
Workload Name . . . . . JES      (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO      (YES or NO)

```

Suppose the regions' service classes are defined as follows (PRODRGNS shown here, TESTRGNS would look the same):

```

Service Class Name . . . . . PRODRGNS (Required)
Description . . . . . Production Regions _____
Workload Name . . . . . STC      (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO      (YES or NO)

```

In this scenario, assume that the regions are running normal, non-conversational transactions. Response time goals are appropriate, and there is enough activity so that WLM can manage the regions as servers virtually all of the time. Transaction AA1 is very important to the business, and you wish to give it both storage and CPU protection.

In this case, protection on a transaction service class level is sufficient. This approach allows you to focus on protecting specific transactions rather than the regions that process them. The protection will be inherited by any regions in which the transactions run, as long as WLM is allowed to manage the region to the transactions' goals. ("Scenario 5" on page 119 shows what happens when WLM is *not* allowed to manage the region to the transactions' goals.)

Assign storage protection to transaction AA1 using the CICS classification rule:

```
Subsystem Type . . : CICS          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type
```

Action	-----Qualifier-----			-----Class-----		Storage
	Type	Name	Start	Service	Report	Critical
___ 1	TN	AA1	___	TRXAA	AA1RPT	YES
___ 1	TN	AA2	___	TRXAA	AA2RPT	NO
___ 1	TN	BB	___	TRXBB	___	NO
___ 1	TN	CC	___	TRXCC	___	NO

Transaction service class TRXAA runs only in the CICSREGP region, and not in CICSREGT; therefore, CICSREGP will inherit the storage protection, and CICSREGT will *not* inherit the storage protection.

Assign CPU protection to the transaction service class TRXAA in the service class definition:

```
Service Class Name . . . . . TRXAA (Required)
Description . . . . . CICS Transactions
Workload Name . . . . . CICS (name or ?)
Base Resource Group . . . . . (name or ?)
Cpu Critical . . . . . YES (YES or NO)
```

Any region serving any TRXAA transactions, even one serving AA2 only, inherits CPU protection. As was true for storage protection, CICSREGT will *not* inherit CPU protection because it does not serve transaction service class TRXAA.

Reporting products which display data about the regions themselves will not show that storage and CPU protection was specified, but will show that they were protected while serving the transactions. (See “Reporting” on page 121.) Service class reports will show the storage and CPU protection assigned to the TRXAA transaction service class.

Scenario 3

In this scenario, again using the CICS regions shown in Figure 25 on page 115, assume that the regions are running non-conversational transactions, but with periods of low activity during which WLM may stop managing them as servers. During this time, it is more likely that the regions' pages will be stolen by competing workloads. In this scenario, assume that transaction BB needs storage and CPU protection.

Protection on a transaction service class level is once again useful, ensuring that the transactions will be protected wherever they run. The regions themselves should also be protected, as WLM may not manage them as servers during the low activity periods.

Assign storage protection to transaction BB using the CICS classification rules:

```
Subsystem Type . . : CICS          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type
```

Action	-----Qualifier-----			-----Class-----		Storage
	Type	Name	Start	Service	Report	Critical
___ 1	TN	AA1	___	TRXAA	AA1RPT	NO
___ 1	TN	AA2	___	TRXAA	AA2RPT	NO
___ 1	TN	BB	___	TRXBB	___	YES
___ 1	TN	CC	___	TRXCC	___	NO

Also, assign storage protection to the regions themselves using the STC classification rules:

```

Subsystem Type . . : STC          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

-----Qualifier-----      -----Class-----      Storage  Manage Region
Action  Type      Name      Start  Service  Report  Critical Using Goals Of:
-----  ---      ---      ---    ---      ---      ---      ---
_____ 1 TN      CICSREGP  _____  PRODRGNS  _____  YES      TRANSACTION
_____ 1 TN      CICSREGT  _____  TESTRGNS  _____  YES      TRANSACTION

```

Assign CPU protection to the transaction service classes TRXBB (the **Cpu Critical** field in the TRXAA and TRXCC service class definitions would remain set to NO):

```

Service Class Name . . . . . TRXBB      (Required)
Description . . . . . CICS Transactions
Workload Name . . . . . CICS          (name or ?)
Base Resource Group . . . . .          (name or ?)
Cpu Critical . . . . . YES           (YES or NO)

```

And also to the regions themselves (PRODRGNS shown here, TESTRGNS would also specify YES in the **Cpu Critical** field):

```

Service Class Name . . . . . PRODRGNS  (Required)
Description . . . . . CICS Regions
Workload Name . . . . . STC          (name or ?)
Base Resource Group . . . . .          (name or ?)
Cpu Critical . . . . . YES           (YES or NO)

```

Note that since both CICSREGP and CICSREGT run transaction BB, *both* regions must be protected.

Reporting products which display data about the regions will show that both CPU and storage protection was specified. (See "Reporting" on page 121.) While the regions are serving transactions, protection will occur if either the regions themselves or any of their served transaction service classes are protected.

Scenario 4

In this scenario, again using the CICS regions shown in Figure 25 on page 115, assume that the regions are running conversational transactions, and response time goals are not appropriate. By exempting the regions from management to the transaction response time goals, the regions will instead be managed according to the goal of the service class assigned to those regions. (If either storage or CPU protection is needed, that goal must be a velocity goal, since discretionary goals are not eligible for storage or CPU protection.) In this scenario, assume that only the production region CICSREGP needs protection.

Assign storage protection to the CICSREGP region. Also, in the same panel, exempt both regions from management to the transaction response time goals:

```

Subsystem Type . . : STC          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

-----Qualifier-----      -----Class-----      Storage  Manage Region
Action  Type      Name      Start  Service  Report  Critical Using Goals Of:
-----  ---      ---      ---    ---      ---      ---      ---
_____ 1 TN      CICSREGP  _____  PRODRGNS  _____  YES      REGION
_____ 1 TN      CICSREGT  _____  TESTRGNS  _____  NO       REGION

```

Assign CPU protection to the PRODRGNS service class (the **Cpu Critical** field in the TESTRGNS service class definitions would remain set to NO):

```

Service Class Name . . . . . PRODRGNS (Required)
Description . . . . . CICS Regions
Workload Name . . . . . STC (name or ?)
Base Resource Group . . . . . (name or ?)
Cpu Critical . . . . . YES (YES or NO)
  
```

Reporting products which display data about the regions will show that CPU or storage protection was specified based on the regions' storage protection value and the CPU protection value of the regions' service classes. (See "Reporting" on page 121.)

Scenario 5

This scenario is similar to "Scenario 2" on page 115, but here you'll see what happens when WLM is not allowed to manage one of the regions to the transactions' goals, and how this will prevent protection of a transaction. In this case, assume that it is transaction BB that you wish to give both storage and CPU protection, and assume that at the same time you have exempted region CICSREGT from management to the transaction response time goals.

You've assigned the storage protection to transaction BB using the CICS classification rule:

```

Subsystem Type . : CICS          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

-----Qualifier-----
Action  Type  Name  Start  Service  Report  Storage
-----  ---  ---  ---  ---  ---  ---
_____ 1  TN   AA1    _____  TRXAA  AA1RPT  NO
_____ 1  TN   AA2    _____  TRXAA  AA2RPT  NO
_____ 1  TN   BB     _____  TRXBB  _____  YES
_____ 1  TN   CC     _____  TRXCC  _____  NO
  
```

You have also assigned CPU protection to the transaction service class TRXBB in the service class definition:

```

Service Class Name . . . . . TRXBB (Required)
Description . . . . . CICS Transactions
Workload Name . . . . . CICS (name or ?)
Base Resource Group . . . . . (name or ?)
Cpu Critical . . . . . YES (YES or NO)
  
```

In the classification rule for CICSREGT, you have exempted the region from management to the transaction response time goals:

```

Subsystem Type . : STC          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

-----Qualifier-----
Action  Type  Name  Start  Service  Report  Storage  Manage Region
-----  ---  ---  ---  ---  ---  ---  Using Goals Of:
_____ 1  TN   CICSREGP  _____  PRODRGNS  _____  YES  TRANSACTION
_____ 1  TN   CICSREGT  _____  TESTRGNS  _____  YES  REGION
  
```

As illustrated in Figure 25 on page 115, transaction BB runs in both regions, CICSREGP and CICSREGT. *Since WLM will not manage region CICSREGT to*

transaction response times, it will not inherit storage or CPU protection from the TRXBB transaction service class. Transaction BB will therefore *not* run with storage or CPU protection in region CICSREGT.

Scenario 6

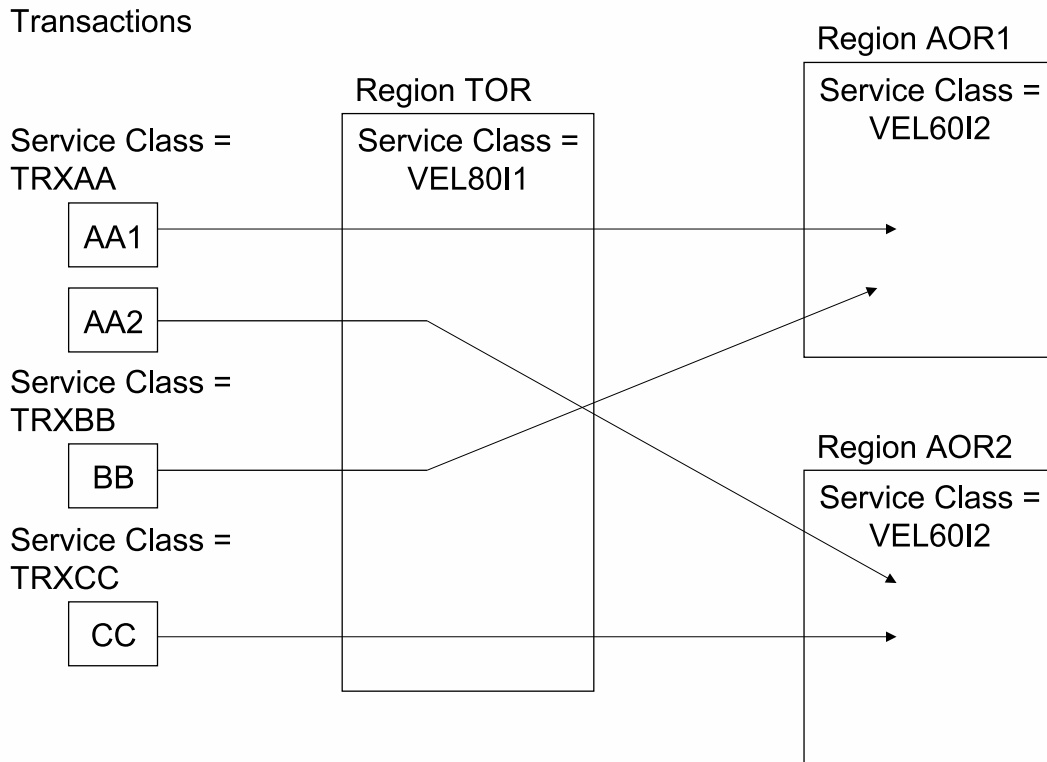


Figure 26. Scenario 6: CICS Regions Adhering to a Work Manager/Consumer Model

Suppose you have the following STC classification rules to classify the CICS regions:

Subsystem Type . . : STC		Fold qualifier names? Y (Y or N)						
Description . . . IBM-defined subsystem type								
Action	Type	Qualifier Name	Start	Class	Report	Storage Critical	Manage Region Using Goals Of:	
___	1	TN	TOR	___	VEL80I1	___	NO	TRANSACTION
___	1	TN	AOR*	___	VEL60I2	___	NO	TRANSACTION

In this scenario, assume that the CICS regions adhere to a Work Manager/Consumer model. This means that the TOR acts as a work receiver of requests from the work originator and as a sender of results back to the work originator. The TOR just distributes the work to consumer processes in the AORs which start application programs to perform the function behind the work requests.

The TOR typically only requires short access to resources, but it also needs very fast access in order to avoid being a bottleneck. The AORs typically run more resource intensive programs which do not require the same fast access to the resources.

In this setup, when contention increases on the system, WLM has no way to give faster access to TORs than to AORs, if no other workload like batch work with lower service goals exists that can be demoted. At higher utilization levels, typically above 85%, a noticeable queue delay within the TORs can be recognized. This reduces the end-to-end response times of the CICS transactions and the throughput of the CICS work.

One could avoid this problem by exempting the CICS regions from being managed towards response time goals:

Subsystem Type . . :		STC		Fold qualifier names? Y (Y or N)			
Description . . .		IBM-defined subsystem type					
		-----Qualifier-----		-----Class-----		Storage	Manage Region
Action	Type	Name	Start	Service	Report	Critical	Using Goals Of:
___ 1	TN	TOR	___	VEL80I1	___	NO	REGION
___ 1	TN	AOR*	___	VEL60I2	___	NO	REGION

The negative effect of this solution is that CICS transaction statistics are no longer available for managing the CICS work and only through classes for reporting purposes. This workaround in fact reduces WLM's capabilities to a reporting only function for CICS.

The best solution to solve this issue is to use option BOTH for the TOR:

Subsystem Type . . :		STC		Fold qualifier names? Y (Y or N)			
Description . . .		IBM-defined subsystem type					
		-----Qualifier-----		-----Class-----		Storage	Manage Region
Action	Type	Name	Start	Service	Report	Critical	Using Goals Of:
___ 1	TN	TOR	___	VEL80I1	___	NO	BOTH
___ 1	TN	AOR*	___	VEL60I2	___	NO	TRANSACTION

This option allows WLM to manage a region by the goals of the regions but also ensures that the region tracks all transaction completions correctly so that WLM can still manage the CICS service classes with response time goals. The new option should only be used for CICS TORs. All AORs should remain at the default (TRANSACTION). In addition, the service class for the CICS TORs should be defined with a higher importance than the service classes for the CICS transactions.

The result is that the CICS TORs now can access the resources they need as fast as possible. Because TORs consume typically only between 5 to 10% of the total CICS resource consumption the goals of the CICS service classes still manage the important parts of the CICS workloads and ensure that work is managed towards response times.

Reporting

Because storage protection can be implicitly applied to an entire transaction service class, and because WLM may or may not be honoring a customer's storage or CPU protection assignment at any given time (for example, due to a RESET), there are seven different "states" that can be reported:

- Storage protection has been explicitly assigned on a classification rule
- Storage protection has been implicitly assigned to a CICS/IMS transaction service class (because it was assigned to at least one transaction in that service class)
- Storage protection is currently being honored
- CPU protection has been explicitly assigned in a service class definition
- CPU protection is currently being honored
- Exemption from transaction response time management has been explicitly assigned on a classification rule.
- Management of a region using the goal of both region and transaction service classes has been explicitly assigned on a classification rule.

These states are reported in SMF type 30 and type 79.1 records. States that apply to an entire service class are also reported in SMF 72.3 records.

Option summary

The following table summarizes the effects of the storage protection, CPU protection, and exemption from transaction response time management options:

Table 9. Summary of options for storage protection, CPU protection, and exemption from transaction response time management

When you...	WLM...
Assign CPU protection to a service class used to manage address spaces and/or enclaves.	Protects any address space or enclave managed according to the goals of that service class. Address spaces being managed as servers are managed according to the goals of the served transactions.
Assign storage protection to an ASCH, JES, OMVS, STC, or TSO address space.	Protects any address space which matches the classification rule, regardless of its server status. Address spaces currently running in multiperiod service classes or in service classes with a short response time goal (20 seconds or less) are excluded from protection.
Assign CPU or storage protection to a CICS or IMS transaction.	Protects any regions recognized as serving that CICS/IMS transaction, unless you prevent WLM from managing the regions as servers. Note that once storage protection is assigned to any transaction in a service class, then <i>all</i> transactions in the same service class become storage protected.
Manage a CICS or IMS region using the goals of the region.	Is prevented from managing the region according to the response time goals of the transactions it is running. It does not recognize the region as a server. The region is managed using the goal of the service class assigned to the region. Transaction response time data is not reported in the service classes to which the transactions are classified, but is still reported in their report classes, if assigned.
Manage a CICS or IMS region using the goals of both region and transaction.	Manages the region using the goal of the service class assigned to the region. This also ensures that the region tracks all transaction completions correctly so that it can still manage the CICS service classes with response time goals. The option should only be used for CICS TORs. All AORs should remain at the default (TRANSACTION). In addition, the service class for the CICS TORs should be defined with a higher importance than the the service class for the CICS transactions.

Table 9. Summary of options for storage protection, CPU protection, and exemption from transaction response time management (continued)

When you...	WLM...
Issue the RESET QUIESCE command.	Will no longer enforce CPU protection. All other options remain unchanged.
Issue the RESET SRVCLASS or RESET RESUME command.	Will assign CPU protection if the target service class has the CPU protection attribute. All other options remain unchanged.

Chapter 13. Defining application environments

An application environment is a group of application functions requested by a client that execute in server address spaces. Workload management can dynamically manage the number of server address spaces to meet the performance goals of the work making the requests. Alternatively, the server address spaces can be started and stopped manually or by automation.

Each application environment should represent a named group of server functions that require access to the same application libraries. Grouping server functions helps simplify library security, application program change control, performance management, and system operation.

For example, an application environment could be one or more DB2 stored procedures. DB2 could have an associated application environment named PAYROLL that handles specific types of stored procedure requests.

Getting started with application environments

The following conditions are required before an application environment can be used:

- The work manager subsystem must have implemented the workload management services that make use of application environments. Examples of IBM-supplied work managers that use application environments are:
 - DB2 (subsystem type DB2)
 - SOMobjects (subsystem type SOM)
 - WebSphere Application Server (subsystem CB)
 - Internet Connection Server, Domino Go Webserver, or IBM HTTP Server Powered by Domino (IHS powered by Domino)
 - MQSeries Workflow (subsystem type MQ)

Refer to subsystem documentation to determine if the subsystems used in your installation make use of application environments.

- One or more application environments must be defined in the workload management service definition. The subsystem reference information should provide guidance for logically grouping applications into application environments.
- The subsystem's work requests must be associated with the appropriate application environment. This step is unique for each subsystem and should be described in the subsystem reference information.

If you request through the service definition that server address spaces be automatically managed, workload management starts and stops server spaces as needed. For example, when a DB2 stored procedure request comes into the system, workload management determines whether there is a server address space to process the work, and if there is, makes the work available to the server. If there is no server address space available, workload management creates one.

Table 10 on page 126 shows the IBM-supplied subsystems that use application environments, the types of requests made by each subsystem, and where the subsystem stores the information that maps the work to application environments.

Table 10. IBM-supplied Subsystems Using Application Environments

Application Environment Subsystem Type	Request Type	Application Environment Mapping
CB	WebSphere Application Server object method requests	Server group name
DB2	Stored procedure requests	DB2 SYSIBM.SYSROUTINES catalog table
IWEB	Hyper Text Transfer Protocol (HTTP) requests	Web configuration file
MQ	MQSeries Workflow requests	MQ process definition for the WLM-managed queue (APPLICID) field
SOM	SOM client object class binding requests	Implementation repository

Specifying application environments to workload management

Note: For applications exploiting the service for defining application environments (IWM4AEDF), you may not need to define the application environment manually. For further information, refer to the appropriate product documentation.

To define an application environment, specify:

- The subsystem type under which the applications are running.
- The JCL procedure to start server address spaces if you wish workload management to automatically manage the number of servers.
- If a JCL procedure is specified, any required start parameters.
- Whether requests can execute in multiple server address spaces and on multiple systems.

Application Environment

Application environment name.

Description (optional)

Description of the application environment.

Subsystem Type

The subsystem type associated with the application environment, such as SOM, DB2, or IWEB.

Procedure Name (optional)

The name of the JCL procedure for starting the address space. The procedure name may be omitted.

Start Parameters (optional)

Parameters required by the JCL procedure to start the address space.

Starting of address spaces

Specify whether workload management can start multiple or single server address spaces for a subsystem instance.

Application Environment

(Required) One to 32 character name of the application environment. You must use this name when specifying to the subsystem how to map work to the

application environments. You also use this name in operator commands when performing actions on the application environment. The name cannot begin with the letters *SYS*.

For guidance in mapping subsystem work to application environments, see the subsystem reference information.

If you check the authorization of application environment server address spaces through a SAF product, such as RACF, or if you plan to do so, choose an application environment name that does not exceed 27 characters. There is a restriction with respect to the maximum length of a server profile name passed to the SAF product (see also “Authorizing application environment servers” on page 135).

Description

(Optional) Up to 32 characters describing the application environment.

Subsystem Type

(Required) Subsystem type is the one to four character name of the subsystem using application environments. This subsystem type is provided to workload management when the subsystem initializes. The types currently in use are listed in Table 10 on page 126. For a subsystem not listed, refer to the subsystem's documentation for the required information.

Note: If you are using DB2 stored procedures, note that the subsystem type DB2 specified here for an application environment is used only for identifying the DB2 subsystem when it begins to use the application environment. There is no connection between this value and classification. For more information on classification, see Chapter 9, “Defining classification rules,” on page 61.

Procedure Name

(Optional) Procedure name is the one to eight character name of the JCL procedure that workload management uses to start a server for the application environment work requests. Refer to the appropriate subsystem documentation for sample JCL procedures to use.

To ensure that an application environment uses the same JCL procedure across the sysplex, either (1) identical procedure proclibs must be maintained across the sysplex or (2) all the procedures must be stored in a single, shared proclib.

If you specify a procedure name, “automatic” control is in effect, and workload management manages the number of servers. If you do not specify a procedure name, manual control is in effect, and servers must be started manually or by automation. In either case, workload management processes work requests from the application environment according to the goals defined for the work once a server address space is available.

Start Parameters

(Optional) Start parameters are the parameters required for the JCL procedure defined in **Procedure Name**. These parameters define how workload management should start the server address spaces. Specify parameters here that you would use for starting a server address space with an MVS START command.

Note that any parameters you specify here override the parameters specified in the JCL procedure for that server.

If you specify the symbol &IWMSSNM, workload management substitutes the subsystem instance name provided to workload management when the subsystem connected to it. Refer to the subsystem reference information to determine the instance name and the appropriate parameters to specify.

Starting of server address spaces for a subsystem instance

(Required) You can limit the number of servers for a subsystem instance.

Reasons for limiting the number of servers might be a need to serialize or limit application environment server activity while testing, or a restriction in the subsystem itself. There are three options:

- Managed by WLM
- Limited to a single server address space per system
- Limited to a single server address space per sysplex

The options that are valid for a subsystem depend on its scope as described in the next section. For guidance on deciding which options to use, and to find out which options are valid for subsystems not explicitly covered in the next section, see the subsystem reference information.

Example of an application environment

To enable workload management to dynamically start server address spaces to process work from a DB2 payroll application environment, do the following:

- Define the application environment

Application Environment

PAYROLL

Description

DB2 Payroll APPLENV

Subsystem Type

DB2

Procedure Name

PAYPROC

Start Parameters

DB2SSN=&IWMSSNM

Server Start

Managed by WLM

When payroll work arrives into the system, workload management manages system resources to meet the goals defined for the work, and dynamically starts and stops server address spaces to process the work.

Selecting server limits for application environments

The previous section describes an option how servers of an application environment can be started. You can allow WLM to manage the number of servers or you can limit them. This option is applied independently for each instance of a subsystem. For IBM-supplied subsystems, you can usually use the default value supplied by the WLM ISPF application. This section defines subsystem instance as it is used by application environments and tells how to select the server start option for IBM-supplied subsystems.

A *subsystem instance* for an application environment is defined as a unique combination of:

- Subsystem type, as specified in the service definition for an application environment, and
- Subsystem name, as defined by the work manager subsystem when it connects to workload management.

A subsystem instance using application environments has one of two different scopes, single-system or sysplex, depending on workload management services used by its subsystem type.

If the scope is single-system, all the server address spaces for the subsystem instance are created on the system where the instance connected to workload management.

If the scope is sysplex, the server spaces can be spread across the sysplex, with workload management starting at most one server on each system. The installation may choose to start additional servers on a system through the START command or automation, and these servers are equally eligible to accept application environment work as the one started automatically.

Note that this scope applies only to application environment server management. A subsystem with single-system scope for application environments, can still perform sysplex-wide functions for other purposes.

You can specify how the number of servers for a subsystem instance will be started when defining the application environment in the service definition. Reasons for limiting the number of servers might be a need to serialize or limit application environment server activity while testing, or a restriction in the subsystem itself. There are three options how to start servers:

- Managed by WLM
- Limited to a single server address space per system
- Limited to a single server address space per sysplex

Options 1 and 2 apply when the subsystem type supports single-system scope.

Options 1 and 3 apply when the subsystem type supports sysplex scope.

“Managed by WLM” for single-system scope means any number of servers may be created for the subsystem instance on the system where it connected to workload management. “Managed by WLM” for sysplex scope means servers may be created for the subsystem instance on any number of systems in the sysplex.

The IBM-supplied subsystems using application environments, their scopes, and valid server limit options are as follows:

Table 11. Application environment server characteristics

Subsystem type	Scope	Valid server limit options
CB	system	<p>1.) Managed by WLM</p> <p>Use this to address the WLM managed servers. This is the default for the WebSphere Application Server.</p> <p>2.) Limited to a single server address space per system</p> <p>Use this if you are testing and want to temporarily limit the number of servers. The WebSphere Application Server itself does not have a requirement for limiting servers.</p>

Table 11. Application environment server characteristics (continued)

Subsystem type	Scope	Valid server limit options
DB2	system	<p>1.) Managed by WLM</p> <p>Use this to address the WLM managed servers. This is the default for DB2.</p> <p>2.) Limited to a single server address space per system</p> <p>Use this if the DB2 stored procedure cannot execute concurrently in multiple address spaces. This option should be used if a stored procedure is to be run in “debug” mode and writes to a trace.</p>
IWEB	system	<p>1.) Managed by WLM</p> <p>Use this to address the WLM managed servers. This is the default for Internet Connection Server, Domino Go Webserver, or IBM HTTP Server Powered by Domino (IHS powered by Domino).</p> <p>2.) Limited to a single server address space per system</p> <p>Use this if you are testing and want to temporarily limit the number of servers. The Internet Connection Server, Domino Go Webserver, or IBM HTTP Server Powered by Domino (IHS powered by Domino) does not have a requirement for limiting servers.</p>
MQ	system	<p>1.) Managed by WLM</p> <p>Use this to address the WLM managed servers. This is the default for MQSeries Workflow.</p> <p>2.) Limited to a single server address space per system</p> <p>Use this if you are testing and want to temporarily limit the number of servers. MQSeries Workflow itself does not have a requirement for limiting servers.</p>
SOM	sysplex	<p>3.) Limited to a single server address space per sysplex</p> <p>SOMobjects requires this limit on the number of servers for an application environment. This option is enforced by the WLM application.</p>

For guidance on deciding which option to use, or to find out what options are valid for other subsystems that use application environments, refer to the subsystem reference information.

How WLM manages servers for an application environment

You allow WLM to manage the number of servers and server address spaces for an application environment if you choose the option “Managed by WLM” for starting servers of a subsystem instance. In this case, WLM has the ability to consider the delays of work requests sent to the application environments in its algorithms to supervise the goal achievement of your service classes and to adjust the resources needed by them. Servers and server address spaces are considered resources which can be made available to the work using an application environment. This is illustrated in the following example:

1. If a DB2 stored procedure request comes into the system, DB2 classifies the work request to WLM and WLM assigns a service class to it. DB2 then queues the work request to the application environment under which the stored procedure should be executed.
2. WLM queues the work requests for each application environment by service class. This allows WLM to understand how the queuing affects the goal achievement of the service class.
 - When the first request is queued to an application environment, workload manager detects that there are no active servers for the request, and automatically starts one.
3. From then on, WLM collects statistics about the queue delays for each application environment and each service class used for the work requests. These queue delays then become part of the WLM algorithms which assess the goal achievement of the service classes and adjust the resources as needed.
 - If, for example, the service class for the DB2 stored procedure requests do not meet their goals, WLM determines which resources are needed to help the work in the service class.
 - If queue delay shows up as the dominating factor, WLM assesses how many additional servers are needed to help the work to meet their goals or at least to see a significant improvement.
 - Once the number of servers has been assessed, WLM makes sure that the system resources are available to start the necessary server address spaces for the required number of servers. During this step, WLM makes sure that more important work is not affected and that the system does not run into a shortage situation because of the new server address spaces. Among the resources considered are:
 - Processor
 - Real storage
 - Auxiliary storage constraints
 - Common storage (SQA) constraints
4. When all tests have completed successfully, WLM starts the required number of server address spaces.

In cases where the system has low utilization, WLM is also able to start one additional server address space for the application environment if this may immediately help the work and it is granted that sufficient resources remain available for other work in the system.

Server address spaces are stopped when the utilization of the servers drops and many servers become idle. WLM then returns the resources used by the server address spaces and thus allows other work to utilize the system.

Using “Managed by WLM” is the optimal way to manage the number of server address spaces. It provides the best performance for the work executed under the application environment and only uses the resources which are optimal based on the goal for the work and the overall utilization of the system.

Using application environments

Application environments can be manually controlled by the installation or automatically controlled by workload management. Note, however, that Dynamic Application Environments can only be automatically controlled by workload management. All of the applications in an application environment are supported

by a single JCL startup procedure. Defining the name of this startup JCL procedure to workload management indicates that workload management should control the server address spaces. This is called automatic control. If you omit the name of the JCL procedure in the application environment definition, then manual control is in effect.

Under manual control, the installation must create and delete, as needed, the server address spaces for each application environment. Note that the **VARY WLM,APPLENV** command can be used to terminate manually started server address spaces (through the quiesce or refresh options), but it will not restart them. For more information on the **VARY WLM,APPLENV** command, see “Using operator commands for application environments” on page 133.

Under automatic control, workload management creates server address spaces as started tasks using the JCL procedure specified in the application environment definitions. The startup parameters may be contained in either the JCL procedure defined for each application environment or in the application environment definition. When the server address spaces are no longer needed, workload management deletes them.

Under automatic control, the quantity of server address spaces is totally controlled by workload management. If an operator or automation starts or cancels the server address spaces under automatic control, workload management will:

- Use servers not started by workload management as if they were started by workload management
- Terminate servers not started by workload management if they are not needed
- Replace a server address space that was unexpectedly cancelled

Note: You should use the **VARY WLM,APPLENV** or **VARY WLM,DYNAPPL** command to manage application environment servers rather than the **CANCEL** command. If there are more than five server cancellations in 10 minutes, workload management stops creating new servers for the application environment. For more information on stop conditions, see “Handling error conditions in application environments” on page 134.

Managing application environments

Once an application environment is defined, and there are server address spaces in use by the subsystem, you can use operator commands to manage the application environment. There are options on the **VARY WLM,APPLENV** or **VARY WLM,DYNAPPL** command that allow you to quiesce, resume, or refresh application environments. These functions allow you, for example, to make changes to the JCL procedure, start parameters, or application libraries, and ensure that new work requests run with the modified information. The resume function also allows you to recover from error conditions that have caused workload management to stop an application environment.

An action taken for an application environment is saved in the WLM couple data set and is not discarded across an IPL. For example, if a quiesce action is in effect and the system is IPLed, the quiesce action remains in effect. You can query the current state of an application environment using the **DISPLAY WLM,APPLENV** or **DISPLAY WLM,DYNAPPLENV** command. The scope of both the **VARY** and **DISPLAY** commands for application environments is sysplex-wide, that is, they affect the application environment on all systems in the sysplex, regardless of the scope of the subsystem using the application environment. The sysplex scope of the

command ensures that an application environment remains consistent across the sysplex, especially where there are shared resources.

This section first introduces the commands that can be used to perform actions on an application environment. Then it describes activities that make use of the commands and describes other conditions that affect the state of an application environment.

Using operator commands for application environments

An application environment initially enters the AVAILABLE state when the service policy that contains its definition is activated. AVAILABLE means the application environment is available for use, and servers are allowed to be started for it. There are three options on the VARY command that you can use to change the state of an application environment after it has been made available:

-

```
VARY WLM,APPLENV=xxxx,QUIESCE or  
VARY WLM,DYNAPPL=xxxx,QUIESCE
```

The quiesce option causes workload management to request the termination of server address spaces for the application environment upon completion of any active requests. Additional work requests are not handled by the servers, although work requests can continue to be queued, waiting for a server. If you do not want work queued, use subsystem functions to stop the queueing.

You can issue a quiesce action for an application environment that is in the AVAILABLE state. When a quiesce action is issued for an application environment, it first enters the QUIESCING state until all servers have been requested to terminate. It then enters the QUIESCED state.

-

```
VARY WLM,APPLENV=xxxx,RESUME or  
VARY WLM,DYNAPPL=xxxx,RESUME
```

The resume option restarts an application environment that was previously quiesced and is in the QUIESCED state. It indicates to workload management that server address spaces can once again be started for this application environment. The new servers process any queued requests and all new requests.

When a resume action is issued for an application environment, it first enters the RESUMING state until all systems in the sysplex have accepted the action. It then enters the AVAILABLE state.

-

```
VARY WLM,APPLENV=xxxx,REFRESH or  
VARY WLM,DYNAPPL=xxxx,REFRESH
```

The refresh option requests the termination of existing server address spaces and starts new ones in their place. Existing servers finish their current work requests and end. The new servers process any queued requests and all new requests.

You can issue a refresh action for an application environment that is in the AVAILABLE state. When a refresh action is issued for an application environment, it first enters the REFRESHING state until all servers have been requested to terminate. It then enters the AVAILABLE state.

Making changes to the application environment servers

The command options described are intended to allow changes to application environments without having to shut down the application itself. Use the quiesce function when you want to do one of the following:

- Perform maintenance on application program libraries statically allocated to server address spaces.
- Update the JCL procedure for an application environment.

When you are ready to put the changes into effect, quiesce the application environment, make the changes to the libraries or service definition as needed, then use the resume function to start new servers with the changed information.

You can also use the quiesce function to suspend execution after repeated application failures or errors. After the errors are corrected, you can resume the application environment.

You may have an application environment where the servers keep application program executable modules in a private cache. If you update the application program, you need to ensure that all copies of the changed modules are replaced wherever they are cached. Use the refresh function to do this.

Changing the definition of an application environment

Workload management initiates a refresh when one of the following changes are made to the application environment definition and activated:

- The JCL procedure name is changed.
- The application environment is switched to automatic control, that is, the JCL procedure name was previously left blank, but now one is provided.
- The server start parameters are changed.
- The limit on server address spaces is changed, for example, from “Managed by WLM” to “Limited to a single address space per system”.

If an application environment is deleted from the service definition, it enters the DELETING state. After workload management requests the termination of all associated servers, the application environment is no longer displayed at all by the **DISPLAY WLM,APPLENV** or **DISPLAY WLM,DYNAPPLENV** command.

Handling error conditions in application environments

Workload management stops the creation of new server address spaces when one of the following conditions exist:

- JCL errors in the procedure associated with the application environment.
- Coding errors in the server code which cause five unexpected terminations of server address spaces within ten minutes.
- Failure of the server address space to connect to workload management due to an invalid invocation environment or invalid parameters.

The application environment first enters the STOPPING state, then the STOPPED state after all systems in the sysplex have accepted the action. In STOPPED state, no new servers are created. Any existing server address spaces continue to process work, and workload management is able to accept new work. If there are no existing servers, then workload management rejects any new work requests.

In STOPPED state, you can make changes to libraries, change the procedure, or make any other changes needed to repair the condition that caused workload

management to stop the application environment. When the problem is resolved, use the resume function to allow workload management to start new servers. The application environment enters the RESUMING state, then the AVAILABLE state after all systems in the sysplex have accepted the action.

Note: If you want to ensure all servers are restarted after a STOPPED state, especially after the JCL procedure or libraries have been modified, you should issue a quiesce function prior to the resume. This ensures there are no servers remaining active that are using back-level information.

Authorizing application environment servers

Because the server address spaces started on behalf of an application environment can run in problem program state, workload management enables you to check the validity of a server through an SAF product such as RACF. When the server is being created, workload management makes an SAF call using a new SERVER class to check whether the server is valid for the application environment. If you do not have the SERVER class defined to your SAF product, workload management allows the server address space to be started.

You can use the SERVER and STARTED classes with a SAF product to restrict access to application environment servers. For example, if you are using DB2, SOM or IWEB servers with application environments, you first associate a userid with the MVS procedure name being used to start the server. This is done using the STARTED resource class or by changing ICHRIN03 (started procedures table). Then you use the SERVER resource class to authorize this userid, and possibly others, to become a server for DB2 stored procedures, SOM method requests, or Internet Connection Server web requests.

Example for restricting access to application environment servers

In this example, the installation has the following situation:

- MVS JCL procedures for DB2 stored procedure servers: PAY1, PAY2, PER1, PER2
These are the JCL procedures that workload management uses to start the DB2 servers that handle stored procedure calls.
- DB2 subsystem names: DB2A and DB2B
These are the subsystem names used when the DB2 subsystem connects to workload management.

1. Activate STARTED and SERVER classes (if not already done):

```
SETR CLASSACT(STARTED) RACLIST(STARTED) GENERIC(STARTED)
SETR CLASSACT(SERVER) RACLIST(SERVER) GENERIC(SERVER)
```

2. Establish an arbitrary user ID to use in a subsequent **RDEFINE** command to tie an MVS procedure name to a server.

```
ADDUSER DB2SERV NOPASSWORD
```

The **NOPASSWORD** keyword here is important, it makes DB2SERV a protected user.

3. Associate the user ID with the started task name.

```
RDEFINE STARTED PAY*.* STDATA(USER(DB2SERV) GROUP(SYS1))
RDEFINE STARTED PER*.* STDATA(USER(DB2SERV) GROUP(SYS1))
```

4. Define server profiles in the form:

```
subsys_type.subsys_name.applenv[.subsys_node]
```

where,

subsys_type

is the subsystem type, as specified in the service definition

subsys_name

is the instance name of the subsystem associated with this server. Refer to subsystem reference information for how to determine the subsystem name. The subsystem uses this name when establishing itself as the work manager for application environment server requests.

subsys_node

is the node name of the server when Work_Manager=Yes is specified. This is an optional parameter.

applenv

is the application environment name, as specified in the service definition

```
RDEFINE SERVER DB2.DB2A.* UACC(NONE)
RDEFINE SERVER DB2.DB2B.* UACC(NONE)
```

Note: The maximum length of a server profile name passed to a SAF product is restricted to 41 characters. WLM cannot start server address spaces for application environments that do not follow this restriction. If you ensure that the applenv is at maximum 27 characters long, it is guaranteed that the maximum server profile name length does not exceed 41 characters.

5. Permit the userid to the servers. This completes the association between the MVS procedure names and the servers:

```
PERMIT DB2.DB2A.* CLASS(SERVER) ID(DB2SERV) ACCESS(READ)
PERMIT DB2.DB2B.* CLASS(SERVER) ID(DB2SERV) ACCESS(READ)
```

6. Refresh the classes to refresh the RACF data base and make these changes go into effect:

```
SETR RACLIST(STARTED) REFRESH
SETR RACLIST(SERVER) REFRESH
```

Chapter 14. Defining scheduling environments

A scheduling environment is a list of resource names along with their required states. It allows you to manage the scheduling of work in an asymmetric sysplex where the systems differ in installed applications, or installed hardware facilities. If an MVS image satisfies all of the requirements in the scheduling environment associated with a given unit of work, then that unit of work can be assigned to that MVS image. If any of the resource requirements are *not* satisfied, then that unit of work *cannot* be assigned to that MVS image.

Scheduling environments and resource names reside in the service definition and apply across the entire sysplex. They are sysplex-oriented. Resource states have a different setting in each system in the sysplex and are, therefore, system-oriented.

Each element in a scheduling environment consists of the name of a resource and a required state of either ON or OFF, as follows:

- If the required state is ON, then the resource state must be set to ON on an MVS image for the requirement to be satisfied.
- If the required state is OFF, then the resource state must be set to OFF on an MVS image for the requirement to be satisfied.

In theory, each resource name represents the potential availability of a resource on an MVS system. That resource can be an actual physical entity such as a data base or a peripheral device, or it can be an intangible quality such as a certain time of day or a certain day of the week. The resource names are abstract, and have no inherent meaning.

For instance, you could define a resource name to be XXXX with a required state of ON. If on system SYS1 the corresponding XXXX resource state is set to ON, then the requirement is satisfied. WLM does not care what "XXXX" means, or whether the ON setting really does signify the existence of some real resource. (You could use XXXX as nothing more than an arbitrary toggle switch, setting it ON for whatever reason you wish.) As long as the settings match, the requirement is satisfied.

This information shows how to define the scheduling environments, the resource names, and their required states. It also shows how to set the resource states on each individual MVS system, and how to associate a scheduling environment name with incoming work.

Getting started with scheduling environments

The following steps are required to use scheduling environments:

- You must define one or more scheduling environments, and all of the resource names and required states that are listed in those scheduling environments, in the workload management service definition. See "Specifying scheduling environments to workload management" on page 138.
- For every system in the sysplex on which you want the resource settings to satisfy either ON or OFF requirements, you must set the individual resource states to either ON or OFF, as appropriate. There is also a third setting, RESET, that satisfies neither an ON nor OFF requirement. See "Managing resource states" on page 139 for more information on the RESET state.

- For each unit of work with resource state requirements that is submitted for execution, you must specify the name of the scheduling environment that should be used to determine which systems can execute that work. See “Associating scheduling environments with incoming work” on page 143.

Specifying scheduling environments to workload management

To define a scheduling environment, you need to specify the following information:

Scheduling Environment Name

(Required) One to 16 character name of the scheduling environment.

- You can have up to 999 unique scheduling environments defined in a service definition.
- Alphanumerics and the special characters @, \$, # and _ are allowed.
- Underscores (_) must be imbedded (for example, PLEX_D01 is valid, but PLEX_ is not).
- Names beginning with SYS_ are reserved for system use.

Description

(Optional) Up to 32 characters describing the scheduling environment.

Once you have defined a scheduling environment, you can start selecting its resource names and required states, as follows:

Resource Name

(Required) One to 16 character name of the resource. There can be more than one resource name listed in a scheduling environment.

- You can have up to 999 unique resource names defined in a service definition.
- Alphanumerics and the special characters @, \$, # and _ are allowed.
- Underscores (_) must be imbedded (for example, PLEX_D01 is valid, but PLEX_ is not).
- Names beginning with SYS_ are reserved for system use.

Resource Description

(Optional) Up to 32 characters describing each resource.

When you select a resource name to become part of the scheduling environment, you also need to specify a required state:

Required State

(Required) For each resource name in a scheduling environment, you must specify a required state of either ON or OFF:

- **ON** specifies that the resource name must be set to ON on a given system for the work associated with this scheduling environment to be assigned to that system.
- **OFF** specifies that the resource name must be set to OFF on a given system for the work associated with this scheduling environment to be assigned to that system.

Scheduling environment example

To define a scheduling environment called DB2LATE that contains the following requirements:

- The “DB2A” resource must be set to ON. (In this example, we’ll say that DB2A has been defined to represent the existence of the DB2 subsystem.)
- The “PRIMETIME” resource must be set to OFF. (In this example, we’ll say that PRIMETIME has been defined to be ON during the normal weekday business hours, and OFF for all other times.)

You would define the following scheduling environment:

Scheduling Environment

DB2LATE

Description

Offshift DB2 Processing

Resource Name	Required State	Resource Description
DB2A	ON	DB2 Subsystem
PRIMETIME	OFF	Peak Business Hours

Null scheduling environments

If you no longer need to restrict where work executes in a sysplex, you can remove all the resource state requirements from a scheduling environment. A null or empty scheduling environment always allows work to be scheduled; that is, any system in the sysplex is satisfactory for work associated with a null scheduling environment. This is a migration aid when you initially have resources that exist on only some of the systems in a sysplex, but later make the resources available to every system. It saves the effort of having to remove the scheduling environment specification from all the incoming work.

Refer to “Working with scheduling environments” on page 222 to see how to use the WLM ISPF application to create and modify scheduling environments.

Managing resource states

For every resource name that is referenced by a scheduling environment, a corresponding resource state must be set on each system in the sysplex. The resource state can be:

- **ON**, which will satisfy a resource state requirement of ON.
- **OFF**, which will satisfy a resource state requirement of OFF.
- **RESET**, which will not satisfy any resource state requirement. Resources are put into the RESET state when:
 - A system is IPLed
 - A policy is activated that defines a resource name that did not exist in the previously active policy

These resource states can be manipulated in three ways:

- The operator command:

```
F WLM,RESOURCE=resource_name,setting
```

where *setting* can be ON, OFF, or RESET.

For example, to set DB2A to ON on system SYS1, here is the command you would enter *on system SYS1*, along with the response you would receive:

F WLM,RESOURCE=DB2A,ON

IWM039I RESOURCE DB2A IS NOW IN THE ON STATE

- The equivalent WLM application programming interface IWMSESET. See *z/OS MVS Programming: Workload Management Services* for more information on using this interface.
- Using SDSF, you can change resource state settings directly on the panel displaying the current states.

Note: Do not attempt to issue the **F WLM,RESOURCE** command from the **COMMNDxx** parmlib member, as this member is processed too early during system initialization. If you want resource states to be set on every system IPL, this needs to be done through an automation product such as System Automation for z/OS as soon as that automation product comes up during system initialization.

It is expected that, in most cases, the mechanics of managing resources states will be handled by installation-provided automation, as opposed to having a human operator issue a modify command every time a resource state is changed. Two examples of how automation could manage resource settings:

- By listening for messages from a subsystem that indicate that subsystem has completed its initialization and is ready to accept work. The automation script could issue the appropriate **F WLM,RESOURCE=subsystem,ON** command on that system. When messages are issued indicating that the subsystem is about to terminate, the script could issue the appropriate **F WLM,RESOURCE=subsystem,OFF** command on that system.
- For time-related resource settings, a simple script can turn settings ON and OFF at certain times of the day (like at the beginning and end of peak business hours).

See *z/OS MVS Programming: Workload Management Services* for more information on automation of resource states, WLM services, and coordination with other job scheduling programs.

When you modify the resource state settings on a given system, you do so *on that system only*. If you modify the DB2A resource state on system SYS1, it has no affect on the DB2A setting on SYS2. If you wish to modify the settings on both systems, you would have to explicitly direct the commands to each individual system.

When *all* of the resource state settings on a particular system match the resource names and required states defined in a particular scheduling environment, only then is that system eligible to receive work associated with the scheduling environment.

Example of resource states

Using the DB2LATE scheduling environment defined in the first example, here's how the resource states might be set, and how that would affect the eligibility of work scheduled with DB2LATE to run on each system in the sysplex.

1. The resource names DB2A and PRIMETIME have just been defined (and not set on the individual systems yet) or the systems have just IPLed:

Resource state	SYS1 settings	SYS2 settings
DB2A	RESET	RESET
PRIMETIME	RESET	RESET

- Because of the existence of DB2 on SYS1 only, the DB2A resource state is modified to ON on that system and OFF on SYS2. Also, automation has been set up to modify the PRIMETIME setting according to the time of day. At the moment, it is 10:00 a.m. on a Monday morning, so PRIMETIME is set to ON on both systems:

Resource state	SYS1 settings	SYS2 settings
DB2A	ON	OFF
PRIMETIME	ON	ON

At the moment, the DB2LATE scheduling environment is not satisfied by the resource state settings of either system. Therefore, any work submitted that is associated with the DB2LATE scheduling environment cannot yet be executed on either system.

- At 5:00 p.m., automation modifies PRIMETIME to OFF on both systems:

Resource state	SYS1 settings	SYS2 settings
DB2A	ON	OFF
PRIMETIME	OFF	OFF

SYS1 now finally has all of the correct resource state settings to satisfy the DB2LATE scheduling environment. The work submitted associated with the DB2LATE scheduling environment can now be executed on SYS1.

In the previous example, work associated with the DB2LATE scheduling environment was assigned to SYS1, because that system satisfied all of the DB2LATE requirements. In the case of a single-system sysplex, this ability to choose the “right” system is no longer applicable. But scheduling environments may still be useful. Consider if, in the previous example, there was only a SYS1 in the sysplex. Only when the DB2A resource setting was set to ON (signalling that DB2 was up and running) and the PRIMETIME resource setting was set to OFF (signalling that the peak business hours were over) would DB2LATE work be processed on SYS1. The DB2LATE scheduling environment would act as a “ready” flag, holding the work until all of the requirements were met.

Figure 27 on page 142 summarizes the relationship between scheduling environments and the resource state settings on several sample systems.

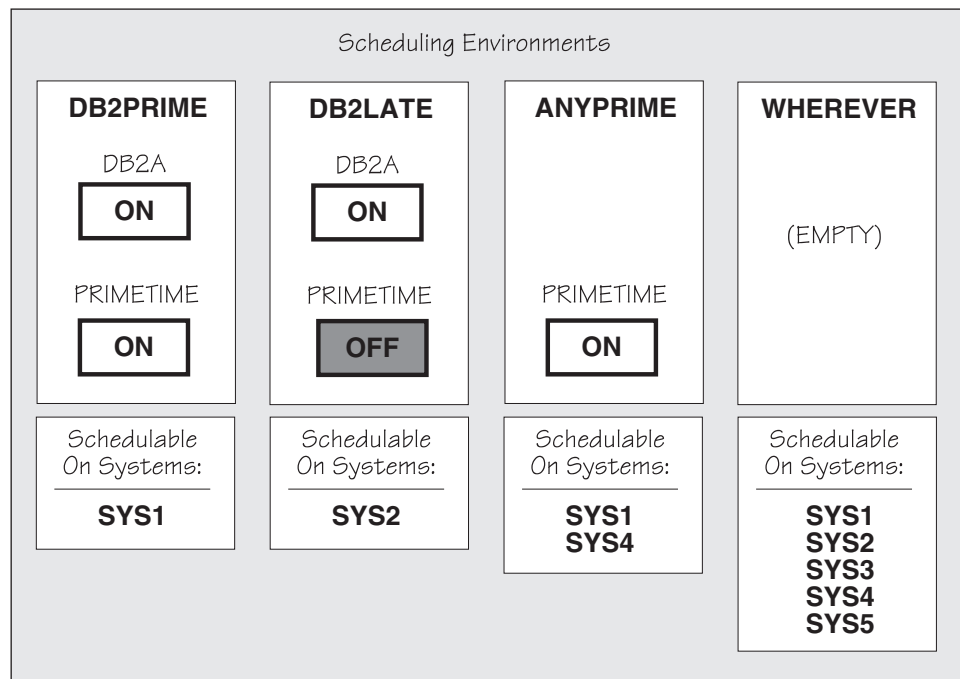
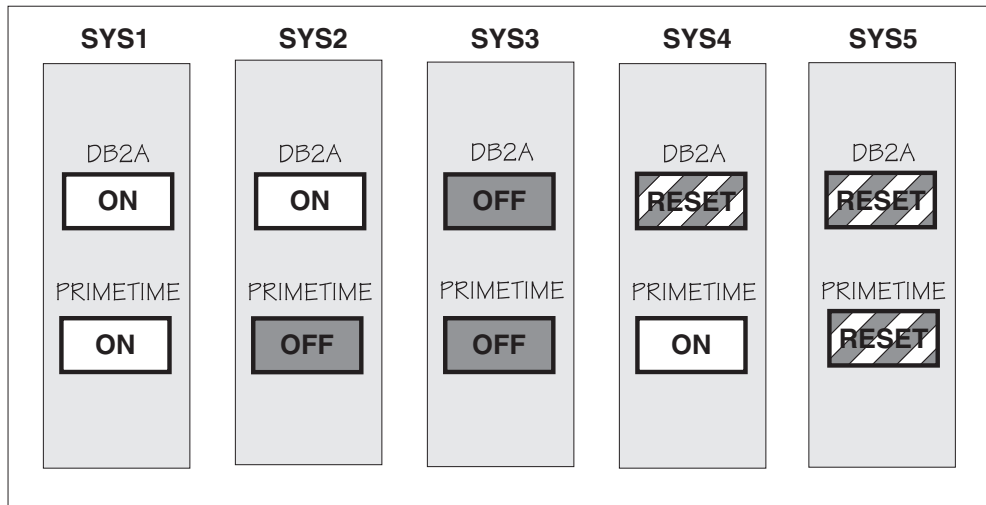


Figure 27. Sample Systems and Scheduling Environments

Note that in Figure 27:

- Work associated with the DB2PRIME scheduling environment can be scheduled only on SYS1, because only that system satisfies both of the requirements.
- Similarly, work associated with the DB2LATE scheduling environment can be scheduled only on SYS2, because only that system satisfies both of the requirements.
- Work associated with the ANYPRIME scheduling environment can be scheduled on either SYS1 or SYS4, because both of those systems satisfy the sole requirement (PRIMETIME must be ON). This scheduling environment does not care about the DB2A setting. Therefore the RESET state for DB2A on SYS4 is irrelevant.

- Work associated with the WHEREVER scheduling environment can be scheduled on any system in the sysplex, because it is empty (it has *no* requirements at all).

Associating scheduling environments with incoming work

Having defined scheduling environments to WLM, and having set the resource states on the individual systems, all you need now is a way to associate the scheduling environment name with actual work.

The **SCHENV** parameter on the JES2 or JES3 JCL JOB statement associates the job with the scheduling environment as shown in the following JCL example:

```
//SHAMIL JOB (C003,6363),'Steve Hamilton',
//          MSGLEVEL=(1,1),
//          REGION=4096K,
//          CLASS=A,
//          SCHENV=DB2LATE,
//          MSGCLASS=0
...

```

This specification associates this batch job with the DB2LATE scheduling environment. It can be coded by the end user, or automatically supplied by installation-provided exits.

If the scheduling environment name specified is not defined in the active WLM policy, the job will fail with a JCL error during conversion, accompanied by an appropriate error message.

Existing JES2 or JES3 exits can be used to change the scheduling environment name associated with batch jobs. This can be done during JCL conversion. These exits can also be used to dynamically generate scheduling environment associations as work is submitted. This could be useful in migrating from another scheduling mechanism to scheduling environments. See *z/OS JES2 Installation Exits* or *z/OS JES3 Customization* for more information.

Displaying information about scheduling environments and resource states

Once you have defined scheduling environments, you can issue several different operator commands, both from MVS and from JES2 or JES3 to display information about the scheduling environments and about the resource states.

MVS operator commands

To display sysplex-level information about a scheduling environment, you can issue the following command from an MVS console:

```
D WLM,SCHENV=scheduling_environment
```

For example, to display information about the DB2LATE scheduling environment, here is the command you would issue and the response you would receive:

```
D WLM,SCHENV=DB2LATE

IWM036I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:           Offshift DB2 Processing
AVAILABLE ON SYSTEMS:  SYS1 SYS3
```

The **AVAILABLE ON SYSTEMS** field shows that at the time this command was issued, only systems SYS1 and SYS3 satisfied the requirements of the DB2LATE scheduling environment.

To display information about *all* scheduling environments in a sysplex, issue the command with an asterisk (*) in the *scheduling_environment* field, as in this example:

```

D WLM,SCHENV=*

IWM036I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:           Offshift DB2 Processing
AVAILABLE ON SYSTEMS:  SYS1 SYS3

SCHEDULING ENVIRONMENT: IMSPRIME
DESCRIPTION:           Primetime IMS Processing
NOT AVAILABLE ON ANY SYSTEM

```

In this example, NOT AVAILABLE ON ANY SYSTEM is shown for the IMSPRIME scheduling environment, meaning that *no* systems in the sysplex currently satisfy the IMSPRIME requirements.

To display system-level information about a scheduling environment, use the **SYSTEM=system_name** parameter. You will see all of the resource names included in that scheduling environment, along with their required and current states. Requirements that are *not* satisfied are marked with an asterisk. So assuming that the DB2LATE scheduling environment is satisfied on SYS1 but not on SYS2, here is what the commands and responses might look like if you wanted the information for both systems:

```

D WLM,SCHENV=DB2LATE,SYSTEM=SYS1

IWM037I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:           Offshift DB2 Processing
SYSTEM:                SYS1
STATUS:                AVAILABLE

RESOURCE NAME   REQUIRED   CURRENT
STATE           STATE
DB2A            ON       ON
PRIMETIME      OFF       OFF

D WLM,SCHENV=DB2LATE,SYSTEM=SYS2

IWM037I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:           Offshift DB2 Processing
SYSTEM:                SYS2
STATUS:                NOT AVAILABLE

RESOURCE NAME   REQUIRED   CURRENT
STATE           STATE
*DB2A          ON       OFF
PRIMETIME      OFF       OFF

```

To display information about a specific resource state on a specific system, use the command:

```
D WLM,RESOURCE=resource_name,SYSTEM=system
```

To display information about the DB2A resource state on system SYS1, for example, here is the command and response:


```
D WLM,RESOURCE=DB2A,SYSTEM=SYS1
```

```
IWM038I 12.21.05 WLM DISPLAY 181
RESOURCE:      DB2A
DESCRIPTION:   DB2 Subsystem
SYSTEM STATE
SYS1          ON
```

To display information about *all* resource settings, use the asterisk in the *resource_name* field. Also, to display information on *all* systems in the sysplex, use the **SYSTEMS** keyword in place of the **SYSTEM=system_name** parameter. So to display information about all the resource states on all systems in the sysplex, here is a typical command and response:

```
D WLM,RESOURCE=*,SYSTEMS
```

```
IWM038I 12.21.05 WLM DISPLAY 181
RESOURCE:      DB2A
DESCRIPTION:   DB2 Subsystem
SYSTEM STATE   SYSTEM STATE   SYSTEM STATE
SYS1          ON          SYS2          OFF          SYS3          RESET

RESOURCE:      PRIMETIME
DESCRIPTION:   Peak Business Hours
SYSTEM STATE   SYSTEM STATE   SYSTEM STATE
SYS1          OFF          SYS2          OFF          SYS3          OFF
```

JES2/JES3 operator commands

The JES2 display command **\$D JOBQ** can be used to list those queued jobs associated with a scheduling environment, as in this example:

```
$djobq,schenv=DB2LATE
```

```
JOB00007 $HASP608 JOB(SHAMIL)
$HASP608 JOB(SHAMIL) STATUS=(AWAITING EXECUTION),CLASS=S,
$HASP608 PRIORITY=1,SYSAFF=(ANY),HOLD=(NONE),
$HASP608 CMDAUTH=(LOCAL),OFFS=( ),SECLABEL=,
$HASP608 USERID=DEALLOC,SPOOL=(VOLUMES=(SPOOL1),
$HASP608 TGS=2,PERCENT=0.3809),ARM_ELEMENT=NO,
$HASP608 SRVCLASS=HOTBATCH,SCHENV=DB2LATE
```

For JES3, use the ***INQUIRY,Q** command, as in this example:

```
*I,Q,SCHENV=IMSPROD
```

```
IAT8674 JOB JOB123 (JOB32787) P=02 CL=Z MAIN(ALLOCATE)
IAT8674 JOB JOBABC (JOB32790) P=02 CL=Z MAIN(ALLOCATE)
IAT8674 JOB JOBDEF (JOB32791) P=02 CL=Z MAIN(ALLOCATE)
IAT8674 JOB JOBGHI (JOB32800) P=02 CL=Z MAIN(ALLOCATE)
IAT8674 JOB JOBJKL (JOB32987) P=02 CL=Z MAIN(ALLOCATE)
IAT8674 JOB JOBMNO (JOB33101) P=02 CL=Z MAIN(ALLOCATE)
```

See *z/OS JES3 Commands* for more information on using JES3 operator commands.

SDSF commands

SDSF can display scheduling environment information, resource information, and allows modification of resource states.

Chapter 15. Managing performance with the guest platform management provider (GPMP)

This information briefly describes the guest platform management provider (GPMP).

Overview of the guest platform management provider (GPMP)

Starting with zEnterprise System, the guest platform management provider (GPMP) is the interface between the z/OS Workload Manager (WLM) and the IBM zEnterprise Unified Resource Manager. The GPMP supports platform management functions and provides policy information to WLM about the platform-wide performance goals of workloads in which the z/OS system is participating.

Within the service definition, you can indicate on which system in the sysplex you want WLM to activate the guest platform management provider. If activation is selected, WLM automatically starts each guest platform management provider and restarts it as needed. Otherwise, you must manually start and stop the guest platform management provider. By default, activation is not enabled.

The guest platform management provider (GPMP) on a z/OS system sends all data over the intranode management network (INMN) using an Open Systems Adapter-Express3 1000BASE-T ethernet adapter (CHPID type OSM). Applications sending or receiving data over the INMN require z/OS Communications Server configuration to access this network. See "TCP/IP in an ensemble" in *z/OS Communications Server IP Configuration Guide* for the steps required to allow TCP/IP to participate in the ensemble, and to allow the guest platform management provider (GPMP) access to the INMN.

For a description of the guest platform management provider commands, refer to *z/OS MVS System Commands*.

For more information about how to activate the guest platform management provider in WLM, refer to Chapter 21, "Using the WLM ISPF application," on page 195.

For information about the guest platform management provider messages, refer to *z/OS MVS System Messages, Vol 9 (IGF-IWM)*.

For information about the hardware requirements of the guest platform management provider and for a detailed description of how to manage performance of zEnterprise systems, refer to the following books:

- *IBM z Systems Ensemble Workload Resource Group Management Guide, GC27-2629*
- *IBM z Systems Ensemble Planning Guide, GC27-2631*

Access to these books requires an IBM Resource Link® user ID and password. To obtain your IBM Resource Link user ID and password, go to the Resource Link home page (<http://www.ibm.com/servers/resourcelink>).

Configuring the guest platform management provider on z/OS

Use this procedure to configure and run the guest platform management provider on your z/OS system. Depending on which z/OS version and release you have installed, the configuration steps you need to perform vary slightly.

Before You Begin

- Determine whether you want to run a guest platform management provider on z/OS and, if so, on which systems. Guest platform management providers are supported on logical partitions (LPARs) and z/VM guests running z/OS V1R10, or higher. See *IBM z Systems Ensemble Planning Guide, GC27-2631* for guidance.
- Make sure your systems have the appropriate level of service to support a guest platform management provider. For all supported releases of z/OS, specific program temporary fixes (PTFs) must be applied before you can configure and start a guest platform management provider. If you are running z/OS as a z/VM guest, specific PTFs are required for z/VM V6.1 as well.

To find instructions about installing the necessary service when migrating to an IBM zEnterprise 196 (z196) server, see the appropriate book:

- For z/OS V1R11 or V1R10 systems, see *z/OS Migration*
- For z/VM V6R1 systems, see *z/VM V6R1 Migration Guide, GC24-6201*.
- Verify that the ensemble network configuration tasks for the intranode management network (INMN) have been completed. To enable the guest platform management provider to send performance data over the INMN, an ensemble administrator or network administrator must configure z/OS Communications Server to access this network. See *z/OS Communications Server IP Configuration Guide, SC31-8775*, for instructions about configuring TCP/IP in an ensemble.
- Make sure that you have the appropriate authority to configure the execution environment for the guest platform management provider on your system. To configure the environment, you must run modified JCL from an authorized user ID with UID=0. IBM provides a sample job, the HVEENV member in SYS1.SAMPLIB, for you to modify and run.
- Make sure that you have the appropriate authority to configure the security environment for the guest platform management provider on your system, or work with your installation's security administrator to accomplish the security-related tasks. This step documents the z/OS Security Server RACF commands that you need to use to control access to specific system resources. To complete this step, you might need to use the *z/OS Security Server RACF Security Administrator's Guide, SA22-7683*.
- Have additional books ready, if you need them. Because you perform all of the tasks in this procedure on your z/OS system, you might need to use these books:
 - *z/OS MVS System Commands* for **DISPLAY WLM** and **MODIFY WLM** command syntax and descriptions.
 - *z/OS MVS System Messages, Vol 5 (EDG-GFS)* for FEW messages that guest platform management providers issue.
 - *z/OS MVS System Messages, Vol 9 (IGF-IWM)* for IWM messages that WLM issues.

About this task

One advantage to using a guest platform management provider is the ability to collect and view application data from IBM middleware products that are

instrumented to the Open Group Application Response Measurement (ARM) 4.0 standard. To collect data from ARM-instrumented applications, z/OS WLM ARM services must be enabled. Steps in this procedure explain how to check the status of and to enable z/OS WLM ARM services.

In a sysplex environment, you can use the WLM ISPF application to enable WLM to start and manage guest platform management providers on all systems that are running supported z/OS releases. You also can name any particular systems that you want to exclude; for those named systems, WLM does not activate the guest platform management provider. Steps in this procedure explain how to enable WLM to manage guest platform management providers automatically, or how to manually start and manage them.

Both z/OS WLM and MODIFY command processing use the started task HVEMCA, in SYS1.PROCLIB, to start the guest platform management provider, which runs in the z/OS UNIX shell environment. After the guest platform management provider starts, it connects to WLM and to the IBM zEnterprise Unified Resource Manager through the INMN.

Procedure

1. Copy the sample JCL from the HVEENV member to your local proclib.
2. In your copy of the JCL from the HVEENV member, modify the variables listed in Table 12 to configure the execution environment for the guest platform management provider. If a dash is shown in the “Suggested value” column, you can use any value.

Table 12. z/OS execution environment variables for the guest platform management provider

Variable Name	Description	Suggested value
DATA-ROOT	Directory on UNIX where the guest platform management provider stores the configuration and diagnostics data.	–
ENV_PROFILE	The UNIX profile for the guest platform management provider to use. This profile contains environmental information, such as TimeZone and Locale.	–
EXTRA_LIBPATH	Optional directory of environment library path.	–
JREBIN_ROOT	Directory on UNIX where the 32-bit Java™ executable is located. Use the latest service level of IBM Runtime Environment for Java 2 (JRE) 1.6 or 1.7.	–
LOGFILE_ROOT	Directory where the guest platform management provider startup-related diagnostics messages are stored.	–
MCA_GROUP	The group created in STEP01 of this JCL.	Use HVEMCA as the group name.
MCA_USER	The user created in STEP01 of this JCL.	Use HVEMCA1 as the user name.

3. In your copy of the JCL from the HVEENV member, add or modify JCL statement parameters to configure the security environment for the guest platform management provider. The HVEENV member in SYS1.SAMPLIB contains sample

z/OS Security Server RACF commands that you or your installation's security administrator can modify to control access to specific system resources.

- a. (Required) Add a group profile and user profile for the HVEMCA started task.

```
ADDGROUP HVEMCA SUPGROUP(SYS1) OWNER(SYS1) OMVS(GID(88888))
ADDUSER HVEMCA1 DFLTGRP(HVEMCA) NOPASSWORD -
        OMVS(UID(0) PROGRAM(/bin/sh) -
        HOME(/usr/lpp/bcp/mca))
RDEFINE STARTED HVEMCA.** -
        STDATA(USER(HVEMCA1) GROUP(HVEMCA))
SETROPTS RACLIST(STARTED) REFRESH
```

- b. (Required) Authorize the HVEMCA started task to use the INMN.

```
SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH) GENERIC(SERVAUTH)
RDEFINE SERVAUTH EZB.OSM.sysname.tcpname
PERMIT EZB.OSM.sysname.tcpname CLASS(SERVAUTH) ID(HVEMCA1) ACCESS(READ)
SETROPTS RACLIST(SERVAUTH) REFRESH
```

- c. If you are using security labels (SECLABELs) in your environment, assign the user of the HVEMCA started task to the security label of SYSMULTI.

```
PERMIT SYSMULTI CLASS(SECLABEL) ID(HVEMCA1) ACCESS(ALTER)
ALTUSER HVEMCA1 SECLABEL(SYSMULTI)
RDEFINE STARTED HVEMCA.** -
        STDATA(USER(HVEMCA1) GROUP(HVEMCA))
        SECLABEL(SYSMULTI)
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS RACLIST(SECLABEL) REFRESH
```

4. Run your copy of the modified JCL from an authorized user with UID=0. The z/OS system issues a message indicating that the job was submitted, and issues another message when the job ends. Check the job log for message FEW2119I, which indicates successful processing of the environment variables.

The job log also contains messages to indicate whether the security commands were successfully processed. For any error messages in the job log, use *z/OS MVS System Messages* to find message descriptions and suggested responses, correct the errors, and rerun the job.

5. From the MVS™ console, issue the DISPLAY WLM command to determine the status of the Application Response Measurement (ARM) services.

```
D WLM,AM,ALL
```

The system responds with message IWM075I. If message IWM075I indicates that ARM services are enabled, proceed to step 6. If message IWM075I indicates that ARM services are disabled, complete the following steps:

- a. Check message IWM075I for started application instances. To collect ARM data for these applications, you must restart them in step 8 after you have successfully configured and started the guest platform management provider.

ARM-instrumented applications must be authorized to use ARM services.

- b. Issue the MODIFY WLM,AM command to enable ARM services.

```
F WLM,AM=ENABLE
```

The system responds with message IWM072I.

6. Edit the z/OS WLM service definition to enable WLM to automatically activate and manage a guest platform management provider on all systems in the sysplex. Using the WLM ISPF application to modify the service definition is the recommended method of managing guest platform management providers. If you do not use this method, you can manually start guest platform management providers by using the MODIFY WLM,GPMP command.

- a. On the Choose Service Definition pop-up, select option 1 or 2 to edit an existing service definition.
 - b. On the Definition Menu, select option 11 to edit the guest platform management provider settings.
 - c. On the GPMP-Settings panel, enter 2 to enable WLM to automatically manage guest platform management providers. On this panel, you also can enter the names of the systems on which you do not want WLM to start a guest platform management provider.
 - d. Return to the Definition Menu.
 - e. Select **File > Save** or **File > Save As** to save your changes to this service definition.
 - f. Select **Utilities > Install definition** to make your changes available for policy activation.
 - g. Select **Utilities > Activate service policy** to activate the policy.
The system responds with messages IWM001I and IWM070I to indicate that the policy has been activated. When the system issues message FEW0600I to indicate successful connection to the INMN, the guest platform management provider is ready to collect and report performance data to the Unified Resource Manager.
7. If you enabled ARM services in step 5 and any ARM-instrumented applications were active when you issued that **MODIFY** command, you must restart these applications so ARM calls can be processed.

Results

The guest platform management provider is correctly configured and ready to collect performance data on your z/OS systems. To verify the status of the guest platform management provider, z/OS WLM, and any ARM-instrumented applications, you can issue the **DISPLAY WLM,AM** command and view the results in the message IWM075I display.

What to do next

If you encounter a problem running the guest platform management provider, IBM Support personnel might request you to run specific commands to collect data required to diagnose the problem.

To collect diagnostic logs and Java dumps, run the **collectLogs.sh** command. This command packages diagnostic data in a Java archive (JAR) file that you can send as directed by IBM Support. The command syntax is:

```
/usr/lpp/bcp/mca/bin/collectLogs.sh -dir path
```

Replace the *path* variable with the directory in which you want to place the JAR file. For example, suppose you issue this command, specifying /tmp as the location for the JAR file. When the **collectLogs.sh** command completes processing, you can see messages similar to the following example:

```
/usr/lpp/bcp/mca/bin/collectLogs.sh -dir /tmp
Debug logs gathered into file /tmp/hvemcaDebug_2010_08_13_09_04_35.jar
PROCESSING COMPLETE
```

Chapter 16. Workload management migration

This information covers two different workload management migration scenarios, depending on whether you already have an existing service definition, or need to create one for the first time:

- **Creating a service definition for the first time**

In this scenario, you do not presently have a service definition, but would like to start exploiting workload management functions.

For the required steps, see “Creating a service definition for the first time.”

- **Migrating to a new z/OS Release with an existing service definition**

In this scenario, you already have a service definition created on a previous release of z/OS and need to accommodate any changes introduced by a higher release.

For the required steps, see “Migrating to a new z/OS release with an existing service definition” on page 155.

This information contains a checklist for each of the two scenarios. The major migration activities are described in detail following the checklist sections, as follows:

- “Restricting access to the WLM service definition” on page 156
- “Start the application and enter/edit the service definition” on page 157
- “Calculate the size of the WLM couple data set” on page 160
- “Allocate a WLM couple data set” on page 160
- “Make a WLM couple data set available to the sysplex for the first time” on page 163
- “Make a newly formatted couple data set available to the sysplex” on page 164
- “Migration considerations for velocity” on page 165
- “Migration considerations for discretionary goal management” on page 166
- “Migration considerations for protection of critical work” on page 166

Creating a service definition for the first time

Use the following checklist to create a service definition for the first time, allocate the WLM couple data set, and then activate a service policy.

1. **Set up performance objectives.** If your installation already has performance objectives, Chapter 4, “Setting up a service definition,” on page 33 is still helpful in setting up a service policy with realistic performance goals.
2. **Set up a service definition** from the performance objectives.
Refer to Chapter 4, “Setting up a service definition,” on page 33 through Chapter 14, “Defining scheduling environments,” on page 137 for further information.
3. **Restrict access to the ISPF administrative application.**
Refer to “Restricting access to the WLM service definition” on page 156 for further information.
4. **Start the application** and enter the service definition.
Refer to “Start the application and enter/edit the service definition” on page 157 for further information.

5. **Upgrade the sysplex couple data set**

Make sure that you have installed the z/OS release and allocated your sysplex couple data set with the IXCL1DSU utility.

For information about how to format the sysplex couple data set, refer to *z/OS MVS Setting Up a Sysplex*.

6. **Allocate a WLM couple data set.**

For more information, refer to “Allocate a WLM couple data set” on page 160.

7. **Make the WLM couple data set available** for use in the sysplex for the first time by either:

- Issuing the **SETXCF** command
- Updating the COUPLExx parmlib member and re-IPLing

For more information, see “Make a WLM couple data set available to the sysplex for the first time” on page 163.

8. **Install a service definition** on the WLM couple data set.

Before you can activate a service policy, you need to install the service definition on the WLM couple data set. To do this, use either the WLM ISPF application, or the Install Definition Utility (new with z/OS R3).

To use the ISPF application, go into it specifying the name of the data set containing your service definition. From the Definition Menu, go to **UTILITIES** on the action bar, then select the pull-down option **Install Service Definition**.

To use the Install Definition Utility, configure the sample JCL (member IWMINSTL, shipped in SYS1.SAMPLIB) as directed in the prolog. Once the JCL has been prepared, it can be started from the command console or submitted as a batch job. A simple service definition has been provided (member IWMSSDEF, also shipped in SYS1.SAMPLIB) that is available for those customers without any other definition.

9. **Adjust SMF recording**

Before you run your systems in goal mode with z/OS V1R3, you should be aware of the changes in your SMF recording. There are several changes to SMF records for goal mode.

In particular, you should turn off SMF type 99 records. They trace the actions SRM takes while in goal mode, and are written frequently. SMF type 99 records are for detailed audit information only. Before you switch your systems into goal mode, make sure you do not write SMF type 99 records unless you want them.

If you do chargeback based on SMF record type 30 or record type 72 records, you may need to update your accounting package.

For more information about SMF record changes for goal mode, see *z/OS MVS System Management Facilities (SMF)*.

10. **Activate a service policy**

Once you have installed a service definition, you can activate a service policy. You can activate a policy either from the administrative application, with the VARY operator command, or the WLM Install Definition Utility.

To activate a service policy from the application, choose the Utilities option from the action bar on the definition menu.

To activate a service policy with the VARY command, specify
VARY WLM,POLICY=xxxx

where xxxx is the name of a policy defined in the installed service definition.

To activate a service policy with the Install Definition Utility, start or submit the sample JCL (member IWMINSTL in SYS1.SAMPLIB).

Once you issue the command, there is an active policy for the sysplex. Systems will start managing system resources to meet the goals defined in the service policy.

For more information about the **VARY** command, see *z/OS MVS System Commands*.

Migrating to a new z/OS release with an existing service definition

Use the following checklist to migrate to the current z/OS release with an existing service definition. You may or may not need to reallocate the WLM couple data set, depending on which release you are migrating from.

1. **Evaluate your service definition.** At some point, either before or after you migrate to the new release, you may need to make one or more adjustments to your service definition.

If you increase your service definition from a lower level to LEVEL011 or above, your service definition is subject to a more rigorous verification. Errors might be flagged if you attempt to either save or install your service definition. The application allows you to save the service definition, but prevents you from installing it until the errors are corrected.

For more information, see the discussion of LEVEL011 in “Service definition functionality levels, CDS format levels, and WLM application levels” on page 158.

2. **Ensure compatibility of downlevel releases.**

If you are running mixed MVS releases on a sysplex, you need to install the compatibility PTFs on all downlevel systems to enable different levels of workload management to coexist until you can upgrade the entire sysplex to the new release. Refer to *z/OS Migration* for further details.

3. **Select the correct z/OS release for policy updates.**

The WLM service definition is stored in ISPF tables, or in XML format. When a new release or a new function APAR adds certain specifications to the service definition, structural changes to the ISPF tables are required. In that case, the WLM application automatically updates the ISPF table structure when you save the service definition in ISPF table format, even if you do not exploit the new functionality. If this occurs, the saved service definition cannot be read by older levels of the WLM application, or the IWMINSTL sample job.

Therefore, it is recommended that you always use XML format when saving service definitions. If using ISPF table format, it is recommended that you start updating a WLM policy with a higher level of administrative application (that is, a higher z/OS release) only when you are sure that you do not have to update that policy data set with a lower level of administrative application in the future. The following releases and new function APARs changed the ISPF table structure:

- OS/390® V1R3, V1R4, V1R6, and V1R10
- z/OS V1R1, V1R8, V1R10, V1R11, and V2R1
- APAR OA47042 for z/OS V2R1 and above
- APAR OA50845 for z/OS V2R1 and above

4. **Start the WLM application.**

For more information, see “Start the application and enter/edit the service definition” on page 157. Please note the discussion of service definition functionality levels in “Service definition functionality levels, CDS format

levels, and WLM application levels” on page 158. Once you choose to use a new functionality level, from that point on you must always use a level of the WLM application that is compatible with that functionality level. It is recommended that you do not use the new functions (which automatically updates the functionality level) until all those systems are upgraded to the new release.

Migration activities

The following sections provide more detail for certain migration activities referenced out of the migration checklists. To determine if you need to perform an activity, refer to the preceding checklists.

Restricting access to the WLM service definition

Before you create a WLM service definition, you should determine who needs access to it, and the kind of functions each person needs to perform. The installation's capacity planners, systems programmers that analyze workloads and the system's performance, system operators, service administrator, and help desk support may all need access to the service definition information.

There are two levels of access you need to consider:

- Access to an MVS data set
- Access to the WLM couple data set

Restricting access to an MVS data set

Any user with access to the administrative application can create their own “practice” service definitions in their own MVS data set. The MVS data set containing the actual service definition that will be installed into the WLM couple data set should be protected. Use a data set profile, just as you would for any other data set. Give READ access only to those people who should be able to view the service definition in the MVS data set, and UPDATE access to those people who should be able to create or modify the service definition in the MVS data set.

Restricting access to the WLM couple data set

Once you have determined who needs access to the WLM couple data set itself, define the kind of access authority required, as follows:

READ With READ access, a user can extract a service definition from the WLM couple data set.

UPDATE

With UPDATE access, a user can:

- Do all the functions available for READ access.
- Install a service definition to a WLM couple data set.
- Activate a service policy.

To control access to the WLM couple data set, use **RDEFINE** to add a profile to the RACF database. Then use **PERMIT** to permit or deny access to the RACF profile. Do not forget to issue the **SETROPTS REFRESH** command after the **PERMIT** command to refresh the RACF data base and activate the changes you have made.

Example of RDEFINE for the WLM couple data set

```
RDEFINE FACILITY MVSADMIN.WLM.POLICY UACC(NONE)
```

Example of PERMIT for the WLM couple data set

```
PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY)
      ID(user)
      ACCESS(READ)
```

```
PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY)
      ID(user)
      ACCESS(UPDATE)
```

where:

user

Indicates the user or user group that needs access to the WLM couple data set.

ACCESS

Indicates the type of access, either READ or UPDATE.

Start the application and enter/edit the service definition

Start the new release's WLM administrative application, and enter your service definition.

If you want to keep the previous release's WLM application, each version of the application must have unique library names. So, when you install the new release, make sure you keep the previous release application on the system under a unique name. You can rename your libraries by using an exit as described in Appendix A, "Customizing the WLM ISPF application," on page 247.

When you enter the service definition, you can keep the service definition in an MVS partitioned data set until you are ready to install the service definition into the WLM couple data set. If you are migrating to use a new version of the WLM application, always save the partitioned data set created by the previous version of the application for backup purposes.

Before starting the WLM application

To start the WLM application, you use the TSO/E REXX exec IWMARIN0. The exec concatenates (via LIBDEF and ALTLIB), the following libraries necessary to run the application:

Table 13. WLM libraries

Library	Content
SYS1.SBLSCLI0	Application REXX code
SYS1.SBLSKEL0	Application skeletons
SYS1.SBLSPNL0	Application panels
SYS1.SBLSTBL0	Application keylists and commands
SYS1.SBLSMSG0	Application messages

The exec also allocates some MVS partitioned data sets for the service definition using TSO ALLOCATE, and then invokes the WLM panels. If you have different data set conventions for your IPCS/WLM libraries, or if you use storage managed data sets, you should use the WLM application exits IWMAREX1 and IWMAREX2. For more information about how to code the exits, see Appendix A, "Customizing the WLM ISPF application," on page 247.

Start the WLM application

To start the application, specify:

```
ex 'SYS1.SBLSCLI0(IWMARIN0)'
```

For more information about IWMARIN0, and for examples on how to start the application specifying the WLM exits, see Appendix A, “Customizing the WLM ISPF application,” on page 247.

Enter/edit the service definition

Type in the service definition (or else edit the existing service definition), and then save it as a partitioned data set (PDS), or a sequential data set (PS). For help on using the application, see Chapter 21, “Using the WLM ISPF application,” on page 195.

Note that if you change the service definition to use certain functions of the new release, you may not be able to use the service definition on the previous release.

Service definition functionality levels, CDS format levels, and WLM application levels

A service definition has a *functionality level* for each release as shown in Table 14:

Table 14. Functionality levels for service definition

Release	Functionality level
OS/390 R4/R5	LEVEL004
OS/390 R6	LEVEL006 or LEVEL007 (LEVEL007 available only with APAR OW33509 installed)
OS/390 R7	LEVEL007 or LEVEL008 (LEVEL008 available only with APAR OW39854 installed)
OS/390 R8	LEVEL007 or LEVEL008 (LEVEL008 available only with APAR OW39854 installed)
OS/390 R9	LEVEL008
OS/390 V1R10 and z/OS V1R1	LEVEL011
z/OS V1R2, V1R3, V1R4, and V1R5	LEVEL013
z/OS V1R6 and V1R7	LEVEL013, LEVEL017 (LEVEL017 available only with APAR OA12784 installed)
z/OS V1R8 and V1R9	LEVEL019
z/OS V1R10	LEVEL021
z/OS V1R11	LEVEL023
z/OS V1R12 and z/OSV1R13	LEVEL025
z/OS V2R1 and z/OS V2R2	LEVEL029, LEVEL030 (LEVEL030 available only with APAR OA47042 installed), LEVEL031 (LEVEL031 available only with APAR OA50845 installed)

Note: LEVEL005, LEVEL009, LEVEL010, LEVEL012, LEVEL014-016, LEVEL018, LEVEL020, LEVEL022, LEVEL024, LEVEL026, LEVEL027, and LEVEL028 are reserved.

If you do not use any of the new functions for a new release, then the functionality level does not change, even if you are using the service definition on a new release. When you install the service definition, the system checks whether you have used any of the new functions, and sets the functionality level. For example, if you created your service definition on z/OS V1R8, then its functionality level is

LEVEL017. If you installed this service definition from a z/OS V1R7 system but did not use any of the new functions, then its functionality level remains LEVEL013.

You should use the new functions when you are comfortable running the new release on your sysplex. Once you use the new functions and increase the functionality level, then you may not be able to use the service definition on a lower level system. For example, you cannot extract a LEVEL017 service definition from a z/OS V1R5 system. You also cannot activate a policy in a LEVEL017 service definition from a z/OS V1R5 system.

The following function, available on z/OS V1R11, increases the service definition level to LEVEL023:

- The number of report classes in the service definition exceeds 999

The following function, available on z/OS V1R12, increases the service definition level to LEVEL025:

- The service definition contains guest platform management provider (GPMP) configuration settings

The following function, available on z/OS V2R1 increases the service definition level to LEVEL029:

- The service definition contains service class(es) assigned to the I/O priority group or I/O priority groups are enabled
- The number of application environments in the service definition exceeds 999
- The service definition contains new qualifier types Client Accounting Information, Client IP Address, Client Transaction Name, Client User ID, or Client Workstation Name in classification rules for subsystem types DB2 or DDF
- The service definition contains groups of the new group types Accounting Information, Correlation Information, Client Accounting Information, Client IP Address, Collection Name, Client Transaction Name, Client Userid, Client Workstation Name, Process Name, Procedure Name, Sysplex Name, Scheduling Environment, Subsystem Parameter, or Subsystem Collection
- The service definition uses a start position for qualifier type Package Name, either in classification rules or group members
- The service definition uses a start position exploiting the new length of 128 bytes for qualifier type Procedure Name, either in classification rules or group members
- The notepad contains more than 500 lines of information.

Note that if you plan to add more than 500 lines of notepad information, you need to re-allocate the WLM couple data set before installing this definition. See “Migration considerations for an increased notepad size” on page 167 for further information.

The following function, available on z/OS V2R1 with APAR OA47042 installed, increases the service definition level to LEVEL030:

- A non-default value is specified for the Reporting Attribute for classification rules.

The following function, available on z/OS V2R1 with APAR OA50845 installed, increases the service definition level to LEVEL031:

- NO is specified for the Honor Priority attribute for any service class on the service class definition panel, or service class override panel.

- A memory limit is specified for any resource group on the resource group definition panel or resource group override panel.

Table 15. The current WLM couple data set format level

Current CDS format level	Description
3	Format updated in OS/390 Release 4, with addition of scheduling environments. This is the format level for OS/390 Release 4 or higher.

Calculate the size of the WLM couple data set

When you reallocate a WLM couple data set, it must be at least as large as the current one or you will not be able to make it available to the sysplex. It is recommended that you increase the size of your couple data set to allow you to exploit new functions, even if you do not plan to exploit new functions immediately.

Use the WLM ISPF Application Utilities option (see “Utilities” on page 204) with the “Allocate couple data set using CDS values” option to determine your couple data set size. If you already have a record of the values, you can skip this step; just ensure you allocate the WLM couple data set with the same or larger values as the current one.

Allocate a WLM couple data set

You need to define a WLM couple data set for storing the service definition information. If you are running a sysplex with mixed release levels, you should format the WLM couple data set from the highest level system. This allows you to use the current level of the WLM application.

To allocate the WLM couple data set you can either use the facility provided in the WLM application, or you can run an XCF utility. For each case, you need to estimate how many workload management objects you are storing on the WLM couple data set. You must provide an approximate number of the following:

- Policies in your service definition
- Workloads in your service definition
- Service classes in your service definition

The values you define are converted to space requirements for the WLM couple data set being allocated. The total space is *not* strictly partitioned according to these values. For most of these values, you can consider the total space to be one large pool.

If you specify 50 service classes, for instance, you are simply requesting that the space required to accommodate 50 service classes be added to the couple data set. You have NOT limited yourself to 50 service classes in the service definition. Although note that if you DO define more than 50 service classes, you will use up space that was allocated for something else.

You should define an alternate WLM couple data set (similar to the sysplex alternate couple data set) for recovery purposes. You can define an alternate WLM couple data set using the same method (either the ISPF application or the XCF utility), but specifying a different data set.

To allocate a WLM couple data set using the ISPF application, choose **Utilities** from the Definition Menu.

To allocate a WLM couple data set using the XCF utility, you can follow some JCL provided in the IWMFTCDS member of SYS1.SAMPLIB.

Sample JCL to allocate a WLM couple data set

To allocate a WLM couple data set, use the sample JCL and fill in the following information:

```
//FMTCDS JOB MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        DEFINEDS SYSPLEX(PLEX1)
                DSN(SYS1.WLMCDS01) VOLSER(TEMPAK)
                MAXSYSTEM(32)
                CATALOG
        DATA TYPE(WLM)
                ITEM NAME(POLICY) NUMBER(10)
                ITEM NAME(WORKLOAD) NUMBER(35)
                ITEM NAME(SRVCLASS) NUMBER(30)
                ITEM NAME(SVDEFEXT) NUMBER(5)
                ITEM NAME(SVDCREXT) NUMBER(5)
                ITEM NAME(APPLENV) NUMBER(50)
                ITEM NAME(SVAEAEXT) NUMBER(5)
                ITEM NAME(SCHENV) NUMBER(50)
                ITEM NAME(SVSEAEXT) NUMBER(5)
```

Where:

SYSPLEX(sysplex)

The name of your sysplex as it appears in your COUPLExx parmlib member.

DSN

The name you are calling your WLM couple data set

VOLSER

A volume that you have access to. If you are using DFSMS, you do not need to specify a VOLSER.

TYPE

The type of function for which this data set is allocated. For a service definition, the type is WLM.

ITEM NAME(POLICY) NUMBER(nn)

Specifies that an increment of space large enough to accommodate the specified number of policies be allocated in the WLM couple data set (Default=5, Minimum=1, Maximum=99).

ITEM NAME(WORKLOAD) NUMBER(nnn)

Specifies that an increment of space large enough to accommodate the specified number of workloads be allocated in the WLM couple data set (Default=32, Minimum=1, Maximum=999).

ITEM NAME(SRVCLASS) NUMBER(nnn)

Specifies that an increment of space large enough to accommodate the specified number of service classes be allocated in the WLM couple data set (Default=128, Minimum=1, Maximum=999).

Note: WLM allows no more than 100 service classes to be defined in a service definition. The default, however, is 128. This will set aside as much space as you will ever need for service classes, as well as a little extra for other WLM objects.

ITEM NAME(SVDEFEXT) NUMBER(nnnn)

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition (IWMSVDEF) be allocated in the WLM couple data set (Default=0, Minimum=0, Maximum=8092).

ITEM NAME(SVDCREXT) NUMBER(nnnn)

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Classification Rules (IWMSVDCR) be allocated in the WLM couple data set (Default=0, Minimum=0, Maximum=8092).

ITEM NAME(APPLENV) NUMBER(nnnn)

Specifies that an increment of space large enough to accommodate the specified number of application environments be allocated in the WLM couple data set (Default=100, Minimum=1, Maximum=3000).

ITEM NAME(SVAEAEXT) NUMBER(nnnn)

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Application Environment Area (IWMSVAEA) be allocated in the WLM couple data set (Default=0, Minimum=0, Maximum=8092).

ITEM NAME(SCHENV) NUMBER(nnn)

Specifies that an increment of space large enough to accommodate the specified number of scheduling environments be allocated in the WLM couple data set (Default=100, Minimum=1, Maximum=999).

ITEM NAME(SVSEAEXT) NUMBER(nnnn)

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Scheduling Environment Area (IWMSVSEA) be allocated in the WLM couple data set (Default=0, Minimum=0, Maximum=8092).

If you encounter a problem during processing, make sure you take a dump by adding the following to your JCL and re-submit.

```
//SYSABEND DD SYSOUT=*
```

Note: The intended users of SVDEFEXT, SVDCREXT, SVAEAEXT, and SVSEAEXT are system management product vendors who wish to include some of their own unique information about customer workload definitions along with the WLM definitions. The WLM interfaces allow these extensions to accompany the service class definitions, report class definitions, or even classification rules. The amount of extra information is specific to each product that exploits these interfaces. That product's documentation should tell the customer how to set SVDEFEXT, SVDCREXT, SVAEAEXT, and SVSEAEXT to ensure that there is sufficient space available in the WLM couple data set to hold the extra information. For more information, see the "Adding Program-Specific Extensions to a Service Definition" topic in the "Using the Administrative Application Services" Chapter in *z/OS MVS Programming: Workload Management Services*.

Increasing the size of the WLM couple data set

You must use a series of **SETXCF** commands to add a new, larger couple data set as the primary WLM couple data set. During this processing you may encounter a message from XCF (IXC250I) indicating your new couple data set is too small. The message indicates which subrecords had insufficient space. If this occurs you must reallocate the new couple data set with a larger size.

From the Definition Menu in the ISPF application, choose **Utilities**. In the Utilities pull-down, choose **4. Allocate couple data set**. Record the values that you see in

that panel. Then choose **5. Allocate couple data set using CDS values**. Record the values that you see in that panel. Now compare the two sets of values and choose the *highest* from each category. For example, if the two panels showed the following values:

Allocate couple data set	Allocate couple data set using CDS values
Service policies . . . _5	Service policies . . . 10
Workloads _40	Workloads _35
Service classes . . . _35	Service classes . . . _30
Application environments 100	Application environments _50
Scheduling environments _80	Scheduling environments _50

Then you should use the highest values in each category, in this case 10, 40, 35, 100, and 80, to allocate your new couple data set.

Make a WLM couple data set available to the sysplex for the first time

This section applies when creating a service definition for the first time. If you already have a service definition and want to make a re-allocated WLM couple data set available, see “Make a newly formatted couple data set available to the sysplex” on page 164.

To make your WLM couple data set available to the sysplex, you must either:

- Update your COUPLExx parmlib member to include the data set name and volume of your WLM couple data set, and re-IPL. Use this option if you have not yet IPLed in a sysplex.
- Issue the **SETXCF** command, if you have already IPLed in a sysplex. You must still update your COUPLExx member for subsequent IPLs.

Using the SETXCF command

To make the WLM couple data set available to the sysplex, you can use the **SETXCF** command. Remember that you still need to update your COUPLExx member as shown in “Updating the COUPLExx member” so that any subsequent IPLs will automatically pick up the WLM couple data sets.

For more information about using the **SETXCF** command, see *z/OS MVS System Commands*.

Examples of the SETXCF command

- To make a primary WLM couple data set called SYS1.WLMCDS01 residing on volume TEMP01 available to the sysplex, enter the following command:

```
SETXCF COUPLE,TYPE=WLM,PCOUPLE=(SYS1.WLMCDS01,TEMP01)
```
- To make an alternate WLM couple data set called SYS1.WLMCDS02 residing on volume TEMP02 available to the sysplex, enter the following command:

```
SETXCF COUPLE,TYPE=WLM,ACOUPLE=(SYS1.WLMCDS02,TEMP02)
```

Updating the COUPLExx member

To make the WLM couple data set available for use in the sysplex, you need to update the DATA keyword in the COUPLExx parmlib member, and IPL so that the member is in use. For more information about updating the COUPLExx member, see *z/OS MVS Setting Up a Sysplex*.

Example of updating the COUPLE_{xx} member

```
DATA      TYPE(WLM)
          PCOUPLE(SYS1.WLMCDS01,TEMP01)
          ACOUPLE(SYS1.WLMCDS02,TEMP02)
```

Where:

TYPE

The function type, WLM.

PCOUPLE(*dataset.name*,*volume*)

The WLM couple data set name, and the volume it resides on.

ACOUPLE(*dataset.name*,*volume*)

The alternate WLM couple data set name, and the volume it resides on. If you do not have an alternate WLM couple data set, then delete this keyword.

Specify the modified COUPLE_{xx} member on your next IPL.

Make a newly formatted couple data set available to the sysplex

This section applies when you already have WLM couple data sets, have just allocated new WLM couple data sets, and want to make them available to the sysplex. You do this, for example, if you want to increase the size of the existing WLM couple data sets.

You must use a series of **SETXCF** command to switch from the currently active primary and alternate couple data sets to the new couple data sets. All systems in the sysplex then operate with the newly allocated data set.

If you are making newly formatted WLM couple data sets available to the sysplex, you can continue to use an older WLM application to modify, install and activate your service definition (as long as new functions are not exploited), or you can switch to using the z/OS Release 2 WLM application.

For more information on compatibility of release levels, WLM application levels, couple data set formats, and functionality levels, see “Service definition functionality levels, CDS format levels, and WLM application levels” on page 158.

Example of making re-allocated couple data sets available

1. Allocate two new couple data sets as described in “Allocate a WLM couple data set” on page 160 For this example, it is assumed you want a primary and an alternate couple data set, and that the names of the new data sets are SYS1.WLMP residing on volume SYS001, and SYS1.WLMA residing on volume SYS002.
2. Make SYS1.WLMP the alternate using the command:

```
SETXCF COUPLE,TYPE=WLM,ACOUPLE=(SYS1.WLMP,SYS001)
```

As part of this processing, **SETXCF** copies the contents of the current primary WLM couple data set to SYS1.WLMP which now is the new alternate.

3. Switch SYS1.WLMP to primary using the command:

```
SETXCF COUPLE,TYPE=WLM,PSWITCH
```
4. Now make SYS1.WLMA the new alternate using the command:

```
SETXCF COUPLE,TYPE=WLM,ACOUPLE=(SYS1.WLMA,SYS002)
```

As in Step 1, this causes the contents of the new primary WLM couple data set SYS1.WLMP to be copied to the new alternate SYS1.WLMA.

Migration considerations for velocity

Initiation delays cause the velocity value to decrease. Recalculate and adjust your velocity goals accordingly. See “Velocity formula” on page 52 for information on calculating velocity.

Before migrating to WLM batch management, you can estimate the new velocity goal for a service class as follows:

Note: All jobs with the same service class should be migrated together to WLM-managed job classes.

- Run the jobs under normal circumstances
- Examine the initiation delay data:
 - In the IWMWRCAA data area, if you are using the workload reporting services:

RCAETOTDQ

Total delay samples, including initiation delay

RCAETOTU

Total using samples

- In the SMF type 72, subtype 3 record, if you are using RMF:

R723CTDQ

Total delay samples, including initiation delay

R723CTOU

Total using samples

Include the initiation delay in the velocity formula for an estimate of the new, lower velocity. Plugging this delay data into the velocity formula gives you:

$$\frac{\text{RCAETOTU}}{\text{RCAETOTU} + \text{RCAETODQ}} \times 100$$

Or:

$$\frac{\text{R732CTOU}}{\text{R723CTOU} + \text{R723CTDQ}} \times 100$$

RMF will do this calculation for you — look for the **INIT MGMT** field in the RMF Monitor I workload activity report (on the line that begins “VELOCITY MIGRATION:”). See “Adjusting velocity goals based on samples included in velocity calculation” on page 56 for more information.

If you had originally given a velocity goal to a service class period *only* because TYPRUN=HOLD time was included in response time goals, you can now give that service class period a response time goal because the TYPRUN=HOLD time is no longer included in the response time. In this case, you no longer need to recalibrate the velocity goal since it has been replaced with the response time goal.

Migration considerations for discretionary goal management

Certain types of work, *when overachieving their goals*, potentially will have their resources “capped” in order to give discretionary work a better chance to run. Specifically, work that is *not* part of a resource group and has one of the following two types of goals will be eligible for this resource donation:

- A velocity goal of 30 or less
- A response time goal of over one minute

Work that is eligible for resource donation may be affected in OS/390 Release 6 and higher if this work has been significantly overachieving its goals. If you have eligible work that must overachieve its goals to provide the required level of service, adjust the goals to more accurately reflect the work's true requirements.

Migration considerations for dynamic alias management

With dynamic alias management, WLM can automatically perform alias address reassignments to help work meet its goals and to minimize IOS queueing.

It is recommended *not* to use dynamic alias management for a device unless all systems sharing that device have dynamic alias management enabled. Otherwise, WLM will be attempting to manage alias assignments without taking into account the activity from the non-participating systems.

See “Specifying dynamic alias management” on page 105 for more information.

Migration considerations for multisystem enclaves

Before using multisystem enclaves, an installation needs to define a specific coupling facility structure named SYSZWLM_WORKUNIT in the CFRM policy. See Chapter 17, “Defining a coupling facility structure for multisystem enclave support,” on page 169 for more information.

Programs that use data from the SMF 30 record may need to be updated in conjunction with multisystem enclave support. The enclave owner's SMF 30 record has new fields containing the CPU time accumulated by all of its split transactions, for *all* systems on which they executed.

For more detailed information on multisystem enclaves, see the “Creating and Using Enclaves” in *z/OS MVS Programming: Workload Management Services*.

Migration considerations for protection of critical work

You should be aware of several options available to help system administrators protect critical work, and how these options may affect other work.

These options include:

- Long-term storage protection
- Long-term CPU protection
- Exemption from management as a transaction server

See Chapter 12, “Defining special protection options for critical work,” on page 109 for more information on these options.

Migration considerations for managing non-enclave work in enclave servers

Starting with z/OS Release V1R12, the non-enclave work of enclave servers is managed towards the first service class period of the address space performance goal. Based on this expanded performance management it is recommended to verify the performance goals for the service class of the address spaces which process enclave work.

For a detailed description about how performance is managed in address spaces with enclaves, see *z/OS MVS Programming: Workload Management Services*.

Migration considerations for an increased notepad size

Starting with z/OS V2R1 the maximum notepad size in the WLM service definition has increased from 500 to 1000 lines. If you plan to use more than 500 lines of notepad information for your WLM service definition, you need to re-allocate the WLM couple data set.

Before you install a service definition with more than 500 lines of notepad information on z/OS V2R1, perform the following steps:

1. Allocate a new WLM couple data set using the IXCL1DSU utility, as described in “Allocate a WLM couple data set” on page 160. Use the current NUMBER specification for each section (POLICY, WORKLOAD, SRVCLASS, etc.).

Note that by using z/OS V2R1 to allocate your WLM couple data set, the allocated space will be sufficient for the increased notepad size. Ensure that the values provided for number of policies, workloads, and service classes are the current values because these values will be used by WLM to calculate the space required. Specifying higher values to allow for growth is acceptable.

2. Switch to the new WLM couple data set or sets, as described in “Make a newly formatted couple data set available to the sysplex” on page 164.
3. Update the COUPLExx parmlib member to specify the new WLM couple data set or couple data sets.

If WLM determines during policy installation that the WLM couple data set is too small to hold the notepad information, the WLM administrative application issues message IWMAM047:

```
WLM couple data set is too small to hold the service definition.
```

It is also possible that message IWMAM044 is issued:

```
Install failed, service definition is not valid.  
Validation reason code: 2903,  
Validation offset: 0.
```

where validation reason code 2903 means “Number of notepad entries (SVNPANPN) exceeds the maximum number allowed (500)”.

If the IWMDINST service is used for policy installation and the WLM couple data set is too small to hold the notepad information, it returns RC 8 with Reason Code xxx083D and VALCHECK_RSN xxx2903.

To resolve this issue, allocate and activate a new, larger WLM couple data set.

For further information refer to “WLM application messages” on page 238 and to *Appendix B. Application Validation Reason Codes* in *z/OS MVS Programming: Workload Management Services*.

WLM managed batch initiator balancing

Starting with z/OS Release V1R4 and JES2 V1R4, and z/OS V1R5 with JES V1R5 WLM is enhanced to improve the balancing of WLM managed batch initiators between systems of a sysplex. While in earlier releases a balancing of initiators between high and low loaded systems was only done when new initiators were started, this is now done when initiators are already available. On highly utilized systems, the number of initiators is reduced while new ones are started on low utilized systems. This enhancement can improve sysplex performance with better use of the processing capability of each system. WLM attempts to distribute the initiators across all members in the sysplex to reduce batch work on highly used systems while taking care that jobs with affinities to specific systems are not hurt by WLM decisions. Initiators are stopped on systems that are utilized over 95% when another system in the sysplex offers the required capacity for such an initiator. WLM also increases the number of initiators more aggressively when a system is low utilized and jobs are waiting for execution.

Batch Initiator Balancing improves the performance and throughput of batch workload over the sysplex. Its intention is not to reach an equally balanced distribution of batch jobs over the LPARs of a sysplex. That is why initiator balancing only comes into effect, when at least one of the systems of the sysplex has a CPU utilization of more than 95%, while other systems have more idle capacity. When the most loaded system has still enough idle capacity to run batch jobs without CPU constraints, it would not improve the total batch throughput if initiators were moved away from that system to other systems, even if the other systems had more idle CPU capacity.

Consider resource group maximum in WLM batch initiator management

Starting with z/OS V1R12 WLM considers the resource group maximum and whether the projected increase in service demand will not exceed the resource group maximum. When the service is already capped due to the resource group maximum, then no additional initiators are started.

Note: This resource maximum check is not done on a LPAR for a service class if WLM has not started an initiator for that particular service class on that LPAR, or if WLM has only started initiators in that service class on other LPARs in the sysplex. In this case, WLM can start an initiator for that service class on that LPAR even if resource group capping is already active for that service class. Otherwise, WLM cannot determine how much capacity is being used on average by a batch job for that service class.

Chapter 17. Defining a coupling facility structure for multisystem enclave support

Some work managers split large transactions across multiple systems in a parallel sysplex, improving the transaction's overall response time. These work managers can use multisystem enclaves to provide consistent management and reporting for these types of transactions.

Among the benefits of using multisystem enclaves:

- All parts of a split transaction are managed to the same service class. If the service class has multiple periods, the CPU usage of the entire transaction is used to switch periods.
- The enclave owner's SMF 30 record includes CPU time accumulated by all of its split transactions, for *all* systems on which they executed.

Before using multisystem enclaves, an installation needs to define a specific coupling facility structure named `SYSZWLM_WORKUNIT` in the CFRM policy. Once the CFRM policy with this structure definition is activated, WLM will automatically connect to the structure, enabling the use of multisystem enclaves.

This information shows how to define the `SYSZWLM_WORKUNIT` structure, a prerequisite to the use of multisystem enclaves. For more information on defining coupling facilities, see *z/OS MVS Setting Up a Sysplex*.

Programs that use data from the SMF 30 record may need to be updated in conjunction with multisystem enclave support. The enclave owner's SMF 30 record has new fields containing the CPU time accumulated by all of its split transactions, for *all* systems on which they executed.

For more detailed information on multisystem enclaves, see the "Creating and Using Enclaves" in *z/OS MVS Programming: Workload Management Services*.

Defining the coupling facility

It may be difficult to size the `SYSZWLM_WORKUNIT` structure at first, as there is no sure way to know exactly how many parallel units-of-work may exist at any given time. The best option is take a best guess at the initial and maximum sizes and then alter the structure size based on performance and/or change in demand.

If the structure's maximum size is defined too low, work managers will experience failures when they try to export enclaves. It is the work manager's responsibility to respond to such a failure. The work requests may instead be run locally (increasing the response time), or the work requests may fail.

The best way to estimate the storage size needed is to use the CFSIZER tool, which you can find at Coupling Facility sizer (<http://www.ibm.com/systems/support/z/cfsizer/>).

Alternately, there are formulas in *PR/SM Planning Guide* to help estimate the storage size needed. As shown in Table 16 on page 170, the TDEC value is the estimated number of concurrently executing parallel units-of-work. Use the TDEC estimate along with the other values explicitly given in the table, as follows:

Table 16. Values to use in storage estimation formulas

Value	Description	Specify:
TDEC	Total directory entry count — the maximum number of concurrently executing parallel units of work	Best estimate
TDAEC	Total data area element count	TDEC X 2
MSC	Maximum number of storage classes	1
MCC	Maximum number of castout classes	1
MDAS	Maximum number of data area elements associated with a directory entry	32
DAEX	Data area element characteristic	3
AAI	Adjunct assignment indicator	0
R_de/R_data	Directory to data ratio	1/2

Once you have estimated the initial and maximum sizes for the SYSZWLM_WORKUNIT structure, define the structure as described in *z/OS MVS Setting Up a Sysplex*. Keep the following points in mind:

- WLM requests a coupling facility with “default” connectivity.
- Non-volatility is not required.
- The coupling facility control code must be at CFLEVEL 9 or higher.

The following sample JCL shows the definition of a SYSZWLM_WORKUNIT structure:

```
//POLICYX JOB ...
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

DATA TYPE(CFRM) REPORT(YES)

DEFINE POLICY NAME(POLICY1) REPLACE(YES)

CF NAME(FACIL01)
TYPE(123456)
MFG(IBM)
PLANT(02)
SEQUENCE(123456789012)
PARTITION(1)
CPCID(00)
SIDE(0)
DUMPSPACE(2000)

CF NAME(FACIL02)
TYPE(123456)
MFG(IBM)
PLANT(02)
SEQUENCE(123456789012)
PARTITION(2)
CPCID(00)
SIDE(1)
DUMPSPACE(2000)

STRUCTURE NAME(SYSZWLM_WORKUNIT)
SIZE(4000)
INITSIZE(3328)
PREFLIST(FACIL02,FACIL01)
```

Shutting down the coupling facility

If it becomes necessary to shut down a coupling facility containing the SYSZWLM_WORKUNIT structure (either to apply maintenance or to reconfigure), there are two options:

- If another coupling facility has enough storage available, use the XES system-managed rebuild function to rebuild the SYSZWLM_WORKUNIT structure into another coupling facility. See *z/OS MVS System Commands* for more information.
- If there is no other coupling facility into which the SYSZWLM_WORKUNIT structure can be rebuilt, the structure will be deleted when its coupling facility is shut down and therefore multisystem enclave support will be disabled (as described in “Coupling facility failures”).

An installation should take the appropriate steps to quiesce any active work which may be using multisystem enclaves before shutting down the coupling facility containing the SYSZWLM_WORKUNIT structure.

Coupling facility failures

If the coupling facility containing the SYSZWLM_WORKUNIT structure fails, or if the structure itself fails, then all existing multisystem enclaves will be lost. It is the work manager's responsibility to respond to such a failure. The work manager may fail the work requests, or it may process them without using multisystem enclaves.

If another coupling facility is available, WLM will automatically create a new (empty) SYSZWLM_WORKUNIT structure in it. New multisystem enclaves can now be created for new work requests.

If the original coupling facility is still intact, but the *link* fails, then the use of multisystem enclaves is temporarily disabled. Again, it is the work manager's responsibility to respond to this situation, either failing the work requests, or processing them without using multisystem enclaves. When the link is restored, then the use of multisystem enclaves can continue.

Chapter 18. The Intelligent Resource Director

The Intelligent Resource Director (IRD) extends the concept of goal-oriented resource management by allowing you to group system images that are resident on the same physical server running in LPAR mode, and in the same Parallel Sysplex[®], into an “LPAR cluster.” This gives workload management the ability to manage processor and channel subsystem resources, not just in one single image but across the entire cluster of system images. Figure 28 shows one LPAR cluster in one central processor complex (CPC):

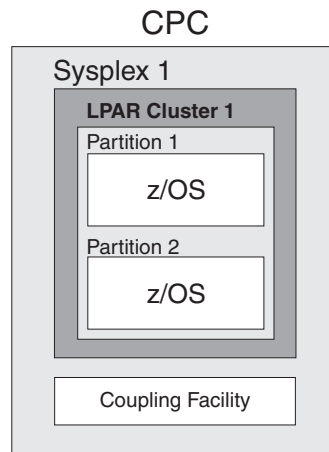


Figure 28. One LPAR cluster on one CPC

A CPC can have multiple LPAR clusters supporting different Parallel Sysplexes, and a Parallel Sysplex can, in turn, comprise multiple LPAR clusters in different CPCs. This is illustrated in Figure 29 on page 174, in which two sysplexes across two CPCs are grouped into four LPAR clusters:

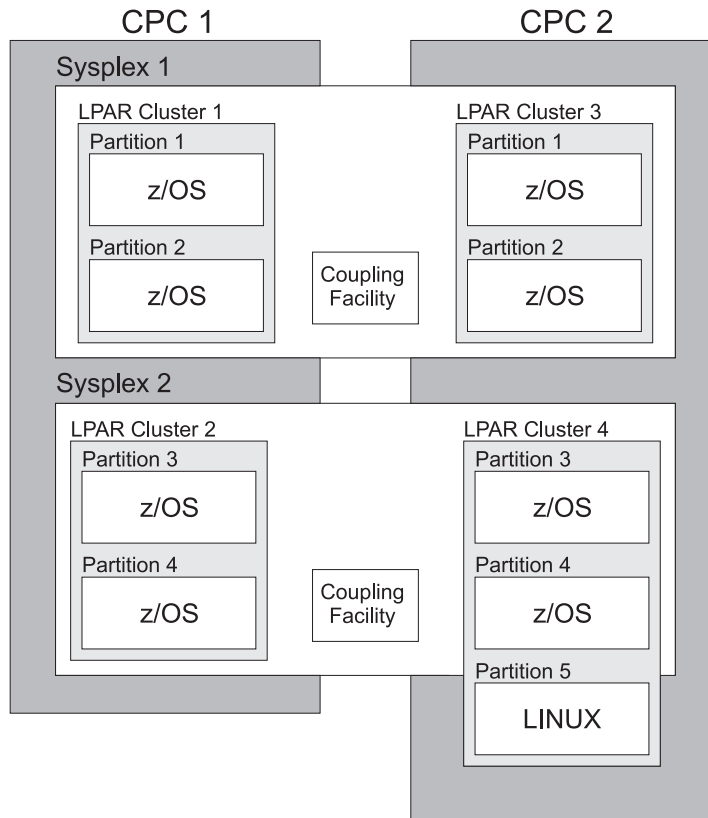


Figure 29. Four LPAR clusters on two CPCs

WLM manages a Parallel Sysplex by directing work to the available resources. With the Intelligent Resource Director, WLM can additionally move resources within an LPAR cluster to the work. Processor resources are automatically moved to the partitions with the greatest need, based on the business goals of the workloads they are running. Channels are automatically moved to the I/O control units with the greatest need, based on the business goals of the workloads using them.

The LINUX partition is not part of the Parallel Sysplex, but WLM manages it as if it were part of it.

The three functions that make up the Intelligent Resource Director are as follows:

- LPAR CPU Management
- Dynamic Channel Path Management
- Channel Subsystem Priority Queuing.

Note: Dynamic channel path management and channel subsystem priority queueing also are functional in single-system environments, such as a z/OS system operating as a monoplex. If a z/OS system is operating as a monoplex, each z/OS system must be assigned a different sysplex name.

LPAR CPU management

LPAR CPU management allows dynamic adjustment of processor resources across logical partitions in the same LPAR cluster. WLM achieves this with two different mechanisms:

- **LPAR weight management**

When you divide your central processing complex into separate logical partitions, each partition is assigned its own *LPAR weight*, which corresponds to the percentage of overall processing power that is guaranteed to the work in that partition. Previously, if the workload shifted to the extent that more processing power was needed in a particular partition, the weights had to be changed manually. With LPAR weight management, you give each logical partition an initial LPAR weight, along with an optional minimum and maximum weight if desired. WLM will then dynamically balance these weights to best meet the goals of the work in the partitions, with no human intervention. Note that the total weight of the cluster as a whole will remain constant, so LPARs outside the cluster are unaffected.

LPAR weight management takes effect when two or more LPARs in the cluster are running CPU-constrained, and the CPC's shared physical CPs are fully utilized. When just one LPAR is using all or most of the CPC because other LPARs are idle, then LPAR weight management will have no effect on resource distribution.

If there are non z/OS partitions in the LPAR cluster, the Intelligent Resource Director manages the weight of these partitions by exchanging weight between the non z/OS and the z/OS partitions of the cluster. LPAR weight of any non z/OS partition will not be reset to its initial weight when the z/OS partition is reset or deactivated. Non z/OS partitions weight can be reset through the service element panel in this situation.

Note: LPAR weight management can be done for standard processors only. LPAR weight management is not supported for zIIPs and zAAPs.

- **VARY CPU management**

VARY CPU Management works for z/OS partitions only. It works hand-in-hand with LPAR weight management. As the LPAR weights change, the number of online logical CPUs is also changed to maintain the best match between logical CPU speed and physical CPU speed. Optimizing the number of logical CPUs benefits workloads that have large amounts of work done under single tasks, and minimizes LPAR overhead for all workloads.

LPAR CPU Management requires System z servers in z/Architecture[®] mode. z/OS images require a CFLEVEL 9 coupling facility structure. Linux for System z requires kernel 2.4 or higher. General purpose CPUs are supported, but Integrated Facility for Linux (IFL) CPUs are not supported.

Dynamic channel path management

Prior to dynamic channel path management, all channel paths to I/O control units had to be statically defined. In the event of a significant shift in workload, the channel path definitions would have to be reevaluated, manually updated via HCD, and activated or PORed into the configuration. Dynamic channel path management lets workload management dynamically move channel paths through the ESCON Director from one I/O control unit to another, in response to changes in the workload requirements. By defining a number of channel paths as *managed*, they become eligible for this dynamic assignment. By moving more bandwidth to

the important work that needs it, your DASD I/O resources are used more efficiently. This may decrease the number of channel paths you need in the first place, and could improve availability — in the event of a hardware failure, another channel could be dynamically moved over to handle the work requests.

Dynamic channel path management operates in two modes: balance mode and goal mode. In balance mode, dynamic channel path management will attempt to equalize performance across all of the managed control units. In goal mode, which is available only when WLM is operating in goal mode on *all* systems in an LPAR cluster, WLM will still attempt to equalize performance, as in balance mode. In addition, when work is failing to meet its performance goals due to I/O delays, WLM will take additional steps to manage the channel bandwidth accordingly, so that important work meets its goals.

Dynamic channel path management requires z/OS and a zSystems server in z/Architecture mode. If a system image running dynamic channel path management in LPAR mode is defined as being part of a multisystem sysplex, it also requires a CFLEVEL 9 or higher coupling facility structure, even if it is the only image currently running on the CPC.

Channel subsystem priority queuing

Channel subsystem priority queuing is an extension of the existing concept of I/O priority queuing. Previously, I/O requests were handled by the channel subsystem on a first-in, first-out basis. This could at times cause high priority work to be delayed behind low priority work. With Channel subsystem priority queuing, if important work is missing its goals due to I/O contention on channels shared with other work, it will be given a higher channel subsystem I/O priority than the less important work. This function goes hand in hand with the dynamic channel path management described — as additional channel paths are moved to control units to help an important workload meet goals, channel subsystem priority queuing ensures that the important workload receives the additional bandwidth before less important workloads that happen to be using the same channel.

WLM sets the priorities using the following basic scheme:

- System related work is given the highest priority
- High importance work missing goals is given a higher priority than other work
- Work meeting goals is managed so that light I/O users will have a higher priority than heavy I/O users
- Discretionary work is given the lowest priority in the system.

Channel subsystem priority queuing requires z/OS and a System z server in z/Architecture mode. It does not require a coupling facility structure.

Example: How the Intelligent Resource Director works

To illustrate how the Intelligent Resource Director works in a mixed workload environment, consider this example:

- You have three workloads running on one server:
 - **Online Transactions**, your most important workload. This runs only during the day shift.
 - **Data Mining**, which has a medium importance. This is always running, and will consume as much resource as you give it.

- **Batch**, which is your lowest importance work. Like data mining, it is always running, and will consume as much resource as you give it.
- In this example, the server is divided into two logical partitions (and both partitions are in the same sysplex):
 - **Partition 1** runs both the online transactions and the batch work, as they happen to share the same database.
 - **Partition 2** runs the data mining work.

Figure 30 shows a day shift configuration. As the online transaction workload is the most important, Partition 1 is given a high enough weight to ensure that the online transaction work does not miss its goals due to CPU delay. Within the partition, the existing workload management function is making sure that the online transaction work is meeting its goals before giving any CPU resource to the batch work.

Day Shift

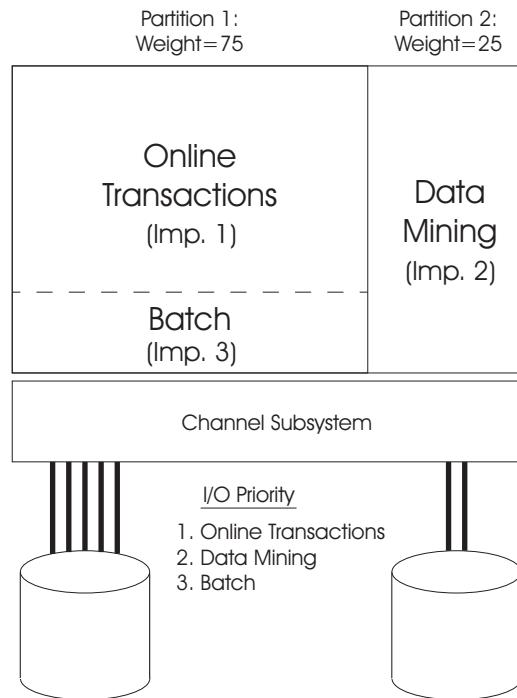


Figure 30. Intelligent Resource Director example – Day shift

The DASD used by the online transaction work is given enough channel bandwidth to ensure that channel path delays do not cause the work to miss its goals. The channel subsystem I/O priority ensures that online transaction I/O requests are handled first. Even though the batch work is running in Partition 1 (with the increased partition weight and channel bandwidth), the data mining I/O requests will still take precedence over the batch I/O requests if the data mining work is not meeting its goals.

Figure 31 on page 178 shows the night shift, when there are no more online transactions. If the partition weights had remained the same, then the batch work would be consuming most of the CPU resource, and using most of the I/O bandwidth, even though the more important data mining work may still be missing its goals. LPAR CPU management automatically adjusts to this change in

workload, adjusting the partition weights accordingly. Now the data mining work will receive the CPU resource it needs to meet its goals. Similarly, dynamic channel path management will move most of the I/O bandwidth back to the data mining work.

Night Shift

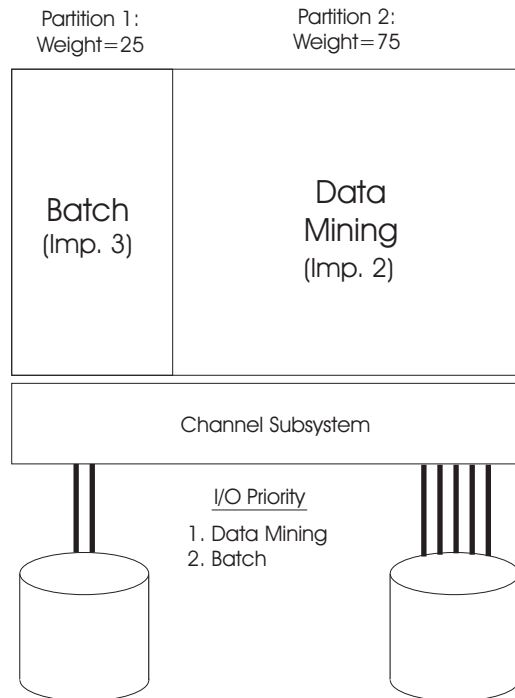


Figure 31. Intelligent Resource Director Example – Night Shift

Making the Intelligent Resource Director work

There are several tasks you need to perform to make the Intelligent Resource Director work in your installation:

- Define the SYSZWLM_XXXXYYYY coupling facility structure.
- Enable LPAR CPU management.
- Enable dynamic channel path management.
- Enable channel subsystem priority queuing.

Defining the SYSZWLM_XXXXYYYY coupling facility structure

Before using either LPAR CPU management or dynamic channel path management in a multisystem sysplex, you'll need to define a specific coupling facility structure named SYSZWLM_XXXXYYYY in the CFRM policy for each LPAR cluster.

In each SYSZWLM_XXXXYYYY structure, the 8-character XXXXYYYY field represents a portion of the CPU ID for the CPC. To obtain this 8-character field, issue the **D M=CPU** command, as follows:

```
SYS1 d m=cpu
SYS1 IEE174I 13.51.42 DISPLAY M 884
PROCESSOR STATUS
ID CPU SERIAL
1 + 1315122064
```

Use the *last* eight characters in the returned CPU ID. So in this example, the coupling facility structure would be named SYSZWLM_15122064.

Define the structure as in the following sample job:

```
//POLICYX JOB ...
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)
DEFINE POLICY NAME(POLICY1) REPLACE(YES)
...
STRUCTURE NAME(SYSZWLM_15122064)
          MINSIZE(10000) /* 1K UNITS */
          INITSIZE(10000)
          SIZE(10000)
          PREFLIST(FACIL01,FACIL02)
```

Activate the new CFRM policy using the **SETXCF START** command, as follows:
SETXCF START,POL,POLNAME=POLICY1,TYPE=CFRM

Once the CFRM policy is activated, WLM will automatically connect to the structure, enabling the use of the Intelligent Resource Director.

Keep the following points in mind:

- WLM requests a coupling facility with “default” connectivity.
- Non-volatility is not required.
- The coupling facility control code must be at CFLEVEL 9 or higher.

If it becomes necessary to shut down a coupling facility containing the SYSZWLM_XXXXYYYY structure (either to apply maintenance or to reconfigure), there are two options:

- If another coupling facility has enough storage available, use the XES system-managed rebuild function to rebuild the SYSZWLM_XXXXYYYY structure into another coupling facility.
- If there is no other coupling facility into which the SYSZWLM_XXXXYYYY structure can be rebuilt, the structure will be deleted when its coupling facility is shut down and therefore LPAR CPU management or dynamic channel path management in a multisystem sysplex will be disabled.

If the coupling facility containing the SYSZWLM_XXXXYYYY structure fails, or if the structure itself fails, then all LPAR CPU management or dynamic channel path management in a multisystem sysplex is stopped. If another coupling facility is available, WLM will automatically create a new (empty) SYSZWLM_XXXXYYYY structure in it. These functions can now resume.

If the original coupling facility is still intact, but the *link* fails, then these functions will be temporarily disabled. When the link is restored, they will resume.

Enabling LPAR CPU management

Once you have defined the SYSZWLM_XXXXYYYY coupling facility structure, the remaining actions you need to take to enable LPAR CPU Management will all occur on the hardware management console (HMC).

- In the Primary Support Element Workplace, make sure that **Not dedicated central processors** is selected in the activation profile of each logical partition. This ensures that the logical partition will use shared CPs instead of dedicated

CPs. On the same panel, specify the initial and reserved number of logical CPs you wish to have available to each logical partition. Production partitions which potentially need access to the full power of the CPC should be defined with the maximum number of logical CPs (equal to the number of shared physical CPs). Other partitions can be defined with a fewer number of logical CPs if they need less power, or if you wish to specifically restrict them to less power. At a minimum, the number of logical CPs should be sufficient to achieve the partition's maximum weight.

For some logical partitions, you may not need to make any of those changes, as the settings are already correct. If you do need to make any of those changes, note that you'll need to deactivate and then reactivate the logical partition for the changes to take effect. For the remaining actions, you will *not* need to do this after making changes.

- For each logical partition that will participate in LPAR weight management, do the following:
 - Make sure that **Initial Capping** is turned off. WLM cannot manage the weight of a logical partition that is capped.
 - Enter the initial processing weight. This becomes the logical partition's weight when it is first IPLed.
 - Enter the minimum and maximum weights. These set the lower and upper limits for the weights that WLM will assign to the logical partition.
 - Check the **WLM Managed** box. This is the final step in activating LPAR weight management.

Once you have made those changes for z/OS images, you do not have to do anything else to activate VARY CPU management. You can disable VARY CPU management, if you wish, by adding the keyword VARYCPU=NO to the IEAOPTxx parmlib member and then issuing the SET OPT=xx command to activate the change. The scope of this command is by single system. If there are multiple systems in the LPAR cluster, then the other systems will continue to use VARY CPU management. You can set VARYCPU=YES to return the system to VARY CPU management. Note that if any CPs were taken offline by an operator, those CPs will need to be configured back online before they can again be managed by VARY CPU management.

Enabling non-z/OS CPU management

In addition to the actions that you have already taken to enable LPAR CPU management, you need to do the following to enable non-z/OS CPU management:

- Before activating the Linux partition, specify the CP management cluster name under the **Options** tab of the activation profile. This causes LPAR to group the Linux partition with the appropriate sysplex.
- For the Linux partition, the system name must be set in order to activate LPAR CPU management. WLM needs the Linux system name in order to handle the Linux partition. This also applies if you use only the PX qualifier. The sysplex name that is specified in the service element panel must be the same as the z/OS sysplex name. The system name must be unique in relation to the other system names in the same sysplex. Set the system name by using the `system_name` attribute in the `/sys/firmware/cpi` directory in sysfs, for example:

```
# echo sysname > /sys/firmware/cpi/system_name
```

where *sysname* is a string that consists of up to 8 characters of the following set: A-Z, 0-9, \$, @, #, and blank.

Use the set attribute to make the setting known to LPAR CPU management, as follows:

```
# echo 1 > /sys/firmware/cpi/set
```

Depending on your Linux distribution, you might be able to configure the `system_name` in the `/etc/sysconfig/cpi` configuration file. VSE/ESA and z/VM[®] set the system name during IPL.

- Goals for non-z/OS partitions are specified in the WLM service definition. You can define velocity goals, but no discretionary goals. Multiple periods are not supported.

Enabling dynamic channel path management

A coupling facility structure is required if you wish to use dynamic channel path management in any logical partition containing a system that is a member of a multisystem complex (even if the system image is the only member of that sysplex on this CPC). You do not need a coupling facility structure if all the logical partitions are running in XCFLOCAL or MONOPLEX mode.

The IBM Redbook, *z/OS Intelligent Resource Director* (SG24-5952), provides extensive guidance on choosing the appropriate channels and control units for dynamic channel path management. Once you have selected them, there are two specific HCD definitions that will need to be changed:

- **Channel definitions**

When defining (or modifying) a channel, you must specify YES in the **Managed** field. You must also specify a dynamic switch (in this case, the ESCON Director) to which the channel is attached. You should also specify the entry switch ID and entry port so that HCD can do consistency checking.

If you are running in LPAR mode, you must also define the name of the sysplex to which a logical partition must belong in order to have this channel in its configuration. Specify this name in the **I/O Cluster** field. You must also define this channel as shared. Note that, unlike traditional shared channels which potentially can be shared by all logical partitions on a CPC, managed channels can only be shared by logical partitions in the same LPAR cluster.

Note: Ensure that each LPAR cluster name (the sysplex name that is associated with the LPAR cluster) is uniquely named across the entire CPC. As managed channels have an affinity to a specific LPAR cluster, non-unique names would create problems with the scope of control.

- **Control unit definitions**

Whereas non-managed channel paths (otherwise called *static* channel paths) are defined in the traditional way, via the CHPID number, a managed path is defined by specifying a double asterisk (**). The number of double asterisks you specify will limit the number of managed channel paths per LPAR cluster. The total number of non-managed and managed channel paths, per CPC, cannot exceed 8.

Important: You must define at least one non-managed channel path (which must be defined as shared) per control unit.

The control unit must be attached to a switch (again, in this case an ESCON Director) which in turn must be attached to managed channels.

After changing the HCD definitions, the remaining actions to enable dynamic channel path management occur in the Primary Support Element Workplace, as follows:

- Ensure that the CPC's reset profile is enabled for dynamic I/O.
- For an LPAR cluster environment, ensure that all partitions in the cluster are authorized to control the I/O configuration.
- Ensure that the automatic input/output (I/O) interface reset option is enabled in the CPC's reset profile. This will allow dynamic channel path management to continue functioning in the event that one participating system image fails.

If you wish to disable dynamic channel path management, issue the **SETIOS DCM=OFF** command. The **SETIOS DCM=ON** command will turn it back on. Issue the **SETIOS DCM=REFRESH** command to refresh the control unit model table (for instance, to include a new **IOST nnn** load module provided by the control unit's manufacturer).

Note: After issuing the **SETIOS DCM=OFF** command, your I/O configuration might now be unable to handle your workload needs, as it will now be in whatever state dynamic channel path management left it before being disabled. You might need to activate a new I/O configuration that will meet your workload needs across the entire LPAR cluster.

Enabling channel subsystem priority queuing

To enable channel subsystem priority queuing, you'll need to do the following:

- In the WLM ISPF application, make sure that **I/O Priority Management** is set to YES on the Service Coefficients/Service Definition panel.
- If your CPC is partitioned, click the **Change LPAR I/O Priority Queuing** icon in the Primary Support Element Workplace. You will see a list of all of the logical partitions. Define the range of I/O priorities that will be used by each image, specifying the minimum and maximum I/O priority numbers. It is recommended that you use a range of eight values (for example 8 to 15), as this will correspond to the number of values in the range that WLM uses. While it is not enforced, it is assumed that you'll set the same range of I/O priorities for all images in the same LPAR cluster. You can, however, prioritize multiple LPAR clusters on the same CPC by setting different ranges for each LPAR cluster.

If you have a partition running a system other than z/OS (for instance z/VM, z/TPF, z/VSE[®]) specify an appropriate default priority by setting the minimum and maximum to the same number. For example, if you have a z/VM partition running work that is equal in importance to the discretionary work in a z/OS partition, you could set the z/VM partition's range to 8–8, and the z/OS partition's range to 8–15. If you have critical OLTP applications running in a partition, on the other hand, you could set the range for that partition to 15–15, and the z/OS partition's range to 7–14. In this way, the OLTP work would always have a higher priority than the z/OS work.

- In the Primary Support Element Workplace, click the **Enable I/O Priority Queuing** icon. You will see a simple panel that reads “Global input/output (I/O) priority queuing.” Click on the **Enable** box. Channel subsystem priority queuing is now enabled for the entire CPC.

For more information

- For more detailed information about the Intelligent Resource Director, see the IBM Redbook, *z/OS Intelligent Resource Director*.
- For more information about defining coupling facilities, see *z/OS MVS Setting Up a Sysplex*.
- For more information about PR/SM, see *PR/SM Planning Guide*.

- For more information about the Primary Support Element Workplace, see *Support Element Operations Guide*.
- For more information about HCD, see *z/OS HCD User's Guide*.
- For more information about RMF, see *z/OS RMF User's Guide*.
- For more information about the IEAOPTxx parmlib member, see *z/OS MVS Initialization and Tuning Reference*.
- For more information about any of the system commands mentioned here, see *z/OS MVS System Commands*.

Chapter 19. Using System z Application Assist Processor (zAAP)

Starting with z/OS V1R6 on z890 and z990 servers, you can run Java applications on a new type of processor called the IBM System z Application Assist Processor (zAAP). You may see this processor also be referred to as IFA (Integrated Facility for Applications) in information related to zAAPs. zAAPs operate asynchronously with the general purpose processors to execute Java programming under control of the IBM Java Virtual Machine (JVM). This helps reduce the demands and capacity requirements on general purpose processors which may then be available for reallocation to other System z workloads. The IBM JVM processing cycles can be executed on the configured zAAPs. zAAPs allow you to integrate and run e-business Java workloads on the same server as your database, helping to simplify and reduce the infrastructure required for web applications.

Benefits of having zAAPs: When running standard CPs on a server, there are a wide variety of speeds or MSU ratings available. The zAAPs, however, are always run at full speed. If you have a workload that is heavy in Java execution, you could run that workload on lower speed standard CPs, along with zAAPs, which would provide significant capacity at a lower cost.

There are no anticipated modifications to Java applications required to use zAAPs.

The steps for starting to use zAAPs and operations considerations are provided in the following. The following tasks relate to using zAAPs:

- Performing capacity planning activities to project how many zAAPs will be needed
- Meeting software and hardware requirements associated with the zAAPs
- Acquiring the zAAPs
- Defining zAAPs to the desired LPARs
- Reviewing parameter settings associated with zAAP usage
- Considering automation changes related to zAAP usage
- Monitoring zAAP utilization and configuring changes appropriately

Performing capacity planning to project how many zAAPs will be needed (zAAP Projection Tool)

Before you have z/OS V1R6, Java SDK 1.4, or a z890 or z990 server, you can do some capacity planning to determine how many zAAPs you need. There is a projection tool (zAAP Projection Tool) available at www.ibm.com/systems/z/hardware/features/zaap/index.html which is a modified Java SDK V1R3, that has some of the same functionality that has been incorporated into Java SDK 1.4 and higher. This tool gathers usage information about how much CPU time is spent executing Java code which could potentially execute on zAAPs. By running a Java workload that is representative of the production system operations, it reports, via the Java log, how much of that workload could be eligible for execution on zAAPs. This information is also useful in predicting the number of zAAPs that might be necessary in order to provide an optimum zAAP configuration.

If you have several systems on the same server on which you are interested in using zAAPs, you can collect the Java log from all the applications running on all the LPARs to have a comprehensive prediction of the total number of zAAPs for a CPC. You may choose to run selected Java workloads, and then extrapolate how much total capacity for zAAPs will be required for all the LPARs where you plan to run Java applications.

Meeting software and hardware requirements associated with the zAAPs

To use zAAPs, there are certain software and hardware requirements. The minimum software requirements are:

- z/OS V1R6 (5694-A01), or later, or z/OS V2R1 (5650-ZOS), or later
- IBM SDK for z/OS, Java 2 Technology Edition V1.4 (Product: 5655-I56, Subscription and Service: 5655-I48) with the PTF for APAR PQ86689, or higher.

With regard to the IBM SDK 1.4 requirement, you should be aware that several products include or require the SDK. The level of SDK that has been included or required, may or may not meet the requirements to use zAAPs. For instance if you are a WebSphere Application Server user, WebSphere Application Server V5.0.2 has included and requires the SDK at the V1R3 level, which does not meet this requirement. However, WebSphere Application Server V5.1 has included and requires the SDK at the 1.4 level which does meet this requirement. It is important to understand which SDK level you are using with your products, and to ensure that you meet the requirements when using zAAPs. Failure to run with the required IBM SDK level means that Java workload will not execute on zAAPs.

The minimum hardware requirements for zAAPs are:

- z890 or z990 server, or later. If you are running on a z990 server, the driver level must be D55, or later.
- The hardware management console (HMC) for defining and configuring the zAAPs must be at driver level D55, or later.

Acquiring the zAAPs

Contact an IBM representative for purchasing zAAPs for your server. zAAPs can only be purchased as an additional Processor Unit (PU) for your server. That is, you cannot convert a standard central processor (CP) you already have acquired to a zAAP.

You may order zAAPs up to the number of permanently purchased CPs, on a given machine model. The number of zAAPs ordered may not exceed the limit of available engines in the machine model.

It is possible to concurrently install temporary capacity by ordering On/Off Capacity on Demand Active zAAPs. The number of On/Off Capacity on Demand zAAPs that you may rent is limited by the number of permanently purchased zAAPs on a given server. On/Off Capacity on Demand zAAPs may not exceed the number of permanently purchased zAAPs on a server.

Defining zAAPs to the desired LPARs

zAAPs are configured via the normal PR/SM logical partition image profile. There are some requirements regarding zAAPs and standard CPs. The following requirements are associated with zAAPs:

- The number of zAAPs must not exceed the number of standard CPs for a server.
- You must have at least one standard CP defined for a partition. The z/OS system needs at least one standard CP online at all times.
- You have the ability to set the number of zAAPs for an LPAR, although you cannot specify a weight or if the zAAPs are dedicated or shared. The zAAPs inherit the dedicated or shared attribute from the standard CPs in that LPAR.
- There is hard capping for zAAPs, but there is no support for soft capping (which is the WLM support for 4-hour rolling average). The existing single set of PR/SM logical partition processor weights (INITIAL, MIN, MAX) are applied independently to the shared standard CPs - capping only applies to shared standard CPs configured to the logical partition. If you use WLM weight management for the LPAR with SCRT, then z/OS WLM will manage shared standard CPs as today, but not the zAAPs.
- zAAPs will not participate in Intelligent Resource Director. The zAAPs will not participate in dynamic share management or in the number of logical zAAPs online.

Note that zAAPs are brought online and offline like standard CPs.

To define the zAAPs to your logical partitions, see *PR/SM Planning Guide*.

Reviewing parameter settings associated with zAAP usage

This section describes the parameter settings associated with zAAP usage.

Review z/OS parameter settings

There are several SRM/WLM options that allow you to control how work is assigned between zAAPs and standard CPs. Java-eligible work that could execute on zAAPs may also execute on standard CPs in order to achieve workload goals.

The IFAHONORPRIORITY statement in parmlib member IEAOPTxx controls the workflow to zAAPs. If you specify IFAHONORPRIORITY=YES (the default) standard CPs can execute both Java™ and non-Java work in priority order, if zAAPs are unable to execute all zAAP-eligible work. IFAHONORPRIORITY=NO requests that standard processors will not process work that is eligible for zAAP processors unless it is necessary to resolve contention for resources with non-zAAP processor eligible work.

If you specify NO for Honor Priority when defining a service class, work in this service class does not receive help from regular CPs when there is insufficient zAAP or zIIP capacity for the workload in the system, regardless of the setting for IFAHONORPRIORITY in parmlib member IEAOPTxx. Regular CPs may still help when it is necessary to resolve contention for resources with regular CP work.

Refer to *z/OS MVS Initialization and Tuning Reference* for more information on the parmlib member IEAOPTxx.

Review Java parameter settings

When running the JVM at SDK 1.4 or higher, there are options provided to allow you to control Java code execution on zAAPs. The default value will cause Java code to be dispatched to zAAPs. The option is specified with other Java startup options. You can use these JVM options to compare execution results of Java code on standard CPs and zAAPs.

-Xifa:on: This is the default. The `-Xifa:on` option indicates to the JVM to call the switch service for zAAPs. If there are no zAAPs available, the JVM will silently disable switching. Once the JVM calls the switch service successfully, it will continue calling the switch service, even if the last zAAP goes offline.

-Xifa:force: The `-Xifa:force` option indicates to the JVM to always call the switching service for zAAPs, even when there are no zAAPs online. As mentioned previously, this option is useful when gathering information for capacity planning.

-Xifa:off: The `-Xifa:off` option indicates to the JVM to bypass all switching to zAAPs. This causes all Java work to be executed on standard CPs.

Refer to *Java Diagnostics Guide* for more information on setting the `-Xifa` startup option in the JVM.

Considering automation changes related to zAAP usage

Based on your IEAOPTxx zAAP settings, there are some automation changes you could make to ensure that Java work will continue to execute in priority order.

Monitoring zAAP utilization and configuring changes appropriately

RMF can monitor zAAP usage. The same skills you have today for monitoring standard CPs can be used for monitoring zAAPs. There are no zAAP-specific skills required for capacity monitoring.

SMF type 30 and type 72 records provide zAAP usage information:

- For the SMF 30 records, both the amount of time zAAP-eligible Java work spends executing on zAAP processors and on standard CPs is reported. Job step time provided by SMF 30 reports the amount of standard CP time consumed by the job step and the amount of zAAP-eligible time consumed by the job step executing Java on standard CPs, if any.

Note that installation exit IEFACTRT allows you to update your system messages with additional accounting information from the SMF type 30 records. The system messages will report information about the zAAP fields: IFA CPU, enclave on IFA, and dep_enclave on IFA.

Refer to *MVS: Installation Exits* for further information about installation exit IEFACTRT.

- For SMF 72 records, the amount of time spent executing on zAAP processors is reported as well as Using and Delay sample counts for zAAP-eligible work.

When running the same Java workload with zAAPs as you were running before without zAAPs, you should expect to see less capacity shown in your SCRT reports (if you are using sub-capacity pricing), as well as less capacity used for standard CPs in your RMF reports. If new Java workload has been added, this increases CP usage.

Refer to *z/OS MVS System Management Facilities (SMF)* for more information on SMF type 30 and type 72 records.

Refer to *z/OS RMF User's Guide* for more information on RMF monitoring.

Note that the diagnosis tools and service aids that you use today, for example, SLIP traps and traces, can be used unchanged with respect to zAAPs.

Chapter 20. Using System z Integrated Information Processor (zIIP)

Starting with V1R8, z/OS on IBM System z9[®] Enterprise Class (z9 EC) or IBM System z9 Business Class (z9 BC) and later servers support the IBM System z Integrated Information Processor (zIIP) — a processor type for a dedicated workload.

Conceptually similar to the System z Application Assist Processor (zAAP), using zIIPs allows to offload certain workloads, for example, selected DB2 tasks, from CPs to zIIPs. This can help free up capacity on general purpose processors which may then be available for reallocation to other System z workloads. With zIIPs available, for example, DB2 can send eligible work to z/OS to be offloaded to zIIPs. Thus, using zIIPs helps to optimize resource usage, contributes to cost-effective System z exploitation and enhances the role of the mainframe as the data hub of the enterprise.

Meeting software and hardware requirements for using zIIPs

To use zIIPs, the minimum software requirements are the following:

- z/OS V1R6 (5694-A01), or later, or z/OS V2R1 (5650-ZOS), or later
- DB2 V8 (5675-DB2) with zIIP enabling APARs installed, or later

To use zIIPs, the minimum hardware requirements are:

- IBM System z9 Enterprise Class (z9 EC) with IBM System z9 Integrated Processor Feature Code 7815, or later family
- IBM System z9 Business Class (z9 BC) with IBM System z9 Integrated Processor Feature Code 7868, or later family

Planning for zIIPs

The SYS1.PARMLIB member IEAOPTxx provides statement **PROJECTCPU**. Specifying the **PROJECTCPU** parameter allows you to project zIIP (and zAAP) consumption when a zIIP (or zAAP) processor is not yet defined to the configuration. RMF and SMF will show the potential calculated zIIP time, so that an accurate zIIP projection can be made. The **PROJECTCPU** parameter can be used while running the target workload, once all software is installed that enables hardware sizing data to be produced.

You can use the **DISPLAY M=CPU** command to show if a zIIP processor is defined in the configuration. (In the **D M=CPU** command output, zIIPs are represented by the letter "I"). A zIIP processor is considered to be defined in the offline or reserved state, as well as in the online state.

See *z/OS MVS Initialization and Tuning Reference* for more information on the parmlib member IEAOPTxx.

The SMF type 30 record (IFASMFR3) includes zIIP consumption fields. See *z/OS MVS System Management Facilities (SMF)* for more information.

The TIMEUSED macro allows zIIP execution time to be requested in addition to the standard CP consumption.

Acquiring zIIPs

Contact an IBM representative for purchasing zIIPs for your IBM System z server. zIIPs can only be purchased as an additional processor unit (PU) for your server. That is, you cannot convert a standard central processor (CP) you already have to zIIP.

You may order zIIPs up to the number of permanently purchased CPs, on a given machine model. The number of zIIPs ordered may not exceed the limit of available engines in the machine model.

It is possible to concurrently install temporary capacity by ordering On/Off Capacity on Demand Active zIIPs. The number of On/Off Capacity on Demand zIIPs that you may rent is limited by the number of permanently purchased zIIPs on a given server. On/Off Capacity on Demand zIIPs may not exceed the number of permanently purchased zIIPs on a server.

Defining zIIPs

zIIPs are configured via the normal PR/SM logical partition image profile.

You can define a logical partition to use one or more zIIPs and/or zAAPs with either of the following combinations:

- One or more dedicated general purpose CPs and one or more dedicated zIIPs/zAAPs
- One or more shared general purpose CPs and one or more shared zIIPs/zAAPs

The mode specified for the logical partition must be set to ESA/390 to allow for the definition of zIIPs or zAAPs to the logical partition.

Futhermore, there are the following requirements regarding zIIPs and standard CPs:

- The number of zIIPs must not exceed the number of standard CPs on a server.
- You must have at least one standard CP defined for a partition. The z/OS system needs at least one standard CP online at all times.
- You can set the number of zIIPs for an LPAR, and each processor pool (for example, CPs and zIIPs) can be assigned a unique weight when the processors are being shared.
- There is hard capping for zIIPs, but there is no support of defined capacity (which is the WLM support for 4-hour rolling average). The existing single set of PR/SM logical partition processor weights (INITIAL, MIN, MAX) are applied independently to the shared standard CPs — capping only applies to shared standard CPs configured to the logical partition. If you use WLM weight management for the LPAR with SCRT, then z/OS WLM manages shared standard CPs as today, but not the zIIPs.
- zIIPs do not participate in Intelligent Resource Director. zIIPs do not participate in dynamic share management or in the number of logical zIIPs online.

Note that zIIPs are brought online and offline like standard CPs.

Combining zIIP-enabled sysplex members with non-zIIP enabled sysplex members is supported.

Reviewing z/OS parameter settings

There are several SRM/WLM options that allow you to control how work is assigned between zIIPs and standard CPs. zIIP-eligible work may also execute on standard CPs in order to achieve workload goals.

Parmlib member IEAOPTxx contains the **IIPHONORPRIORITY** statement which controls the workflow to zIIPs. If you specify IIPHONORPRIORITY=YES this indicates that standard CPs may execute zIIP-eligible and non-zIIP-eligible work in priority order, if zIIP processors are unable to execute all zIIP-eligible work. This is the default.

Specifying IIPHONORPRIORITY=NO means that standard processors will not process zIIP processor eligible work *unless it is necessary* to resolve contention for resources with non-zIIP processor eligible work.

If you specify NO for Honor Priority when defining a service class, work in this service class does not receive help from regular CPs when there is insufficient zAAP or zIIP capacity for the workload in the system, regardless of the setting for IIPHONORPRIORITY in parmli member IEAOPTxx. Regular CPs may still help when it is necessary to resolve contention for resources with regular CP work.

See *z/OS MVS Initialization and Tuning Reference* for more information on the parmli member IEAOPTxx.

Using zIIPs — miscellaneous services

The WLM IWMEQTME and IWM4EDEL services are enhanced to support zIIP usage. For further information about the new parameters of IWMEQTME and IWM4EDEL, refer to *z/OS MVS Programming: Workload Management Services* .

Check RMF measurements for zIIP usage statistics. Refer to *z/OS RMF Report Analysis* for further information on RMF's reporting of zIIP.

Activating zIIPs

z/OS zIIP support is operational when the SMP/E installation is complete and the target image has been restarted. The zIIP engines can be configured online any time after restarting the logical partition. No further customization is required.

Chapter 21. Using the WLM ISPF application

This information explains how to use the ISPF application. It explains the functions that are available, and how you can navigate through the panels.

Before you begin

You should prepare at least one service policy and your classification rules to be ready to start using the ISPF application. Your service policies and classification rules make up a service definition. You can store a service definition in the following kinds of data sets:

WLM couple data set

In order for all systems in a sysplex to process with an active service policy, they must all be able to access a service policy. They all access the policy from a WLM couple data set. To use workload management, you must allocate a WLM couple data set, define it to the sysplex, and install your service definition onto it. You can allocate the WLM couple data set from the application. Only one service definition can be installed on the WLM couple data set.

MVS partitioned data set (PDS)

You do not need to preallocate the data sets. You specify a data set name, and the application allocates it for you. You can save one service definition per MVS PDS.

Notes:

1. If you use customized data sets in your installation, or if you use DFSMS, you can use WLM application exits IWMAREX1 and IWMAREX2 to specify those changes. See Appendix A, “Customizing the WLM ISPF application,” on page 247 for how to code the exits.
2. The data set *userid.WLM.SAVExx* (where *userid* is the TSO ID running the application and *xx* is some numeric value such as SAVE01) is allocated by the WLM application for recovery and is deleted by WLM upon exiting the application. This naming convention should therefore *not* be used for a new service definition.

MVS sequential data set (PS)

You can store a service definition in XML format in a sequential data set. You need not preallocate the sequential data sets. Specify a data set name, then the application allocates it for you.

Notes:

1. If you use customized data sets in your installation, or if you use DFSMS, you can use WLM application exits IWMAREX1 and IWMAREX2 to specify those changes. See Appendix A, “Customizing the WLM ISPF application,” on page 247 for how to code the exits.
2. The data set *userid.VDEF.TEMP.Ddddddd.Tttttt* (where *userid* is the TSO ID running the application, *dddddd* is the current date and *ttttt* is the current time) is allocated by the WLM application to temporarily save the XML service definition as ISPF tables during editing. The data set is deleted by WLM upon exiting the application. The naming convention should therefore *not* be used for a new service definition.

Panel areas and how to use them

Most panels have a menu bar, action field, status line, scrollable area, function key area, and command line. You tell the application what actions to perform by making choices or typing information on a panel.

In this topic, examples of panels and pop-ups are shown to help familiarize you with the product. The examples closely match what you see on your terminal, but in some cases the spacing or function key settings may not exactly match what you see on your terminal.

Using the menu bar

A menu bar at the top of every panel shows the actions you can take on that panel. Press F10, or the Home key on some terminals, to move the cursor to the beginning of the menu bar from any position on a panel.

Figure 32 shows an example of the menu bar on the Definition menu.

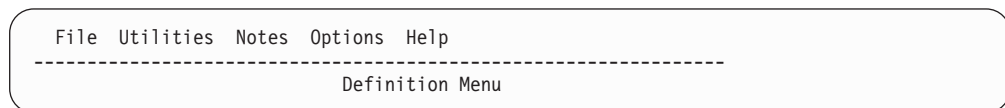


Figure 32. Menu Bar on the Definition Menu

To select an action, use the Tab or cursor movement keys to position the cursor on your choice, then press ENTER.

When you select an option on the menu bar, WLM displays a pull-down with choices related to the option you selected. While a pull-down is displayed, only the actions in the pull-down or on the menu bar are available. For example, if you select a pull-down and the option you want is not listed in it, you can select another pull-down on the menu bar.

To select an option in a pull-down, type the number of your choice in the action field and press ENTER. You can also use the cursor movement keys to position the cursor on your choice, then press ENTER. Figure 33 shows an example of the pull-down choices on the File option on the Definition Menu.

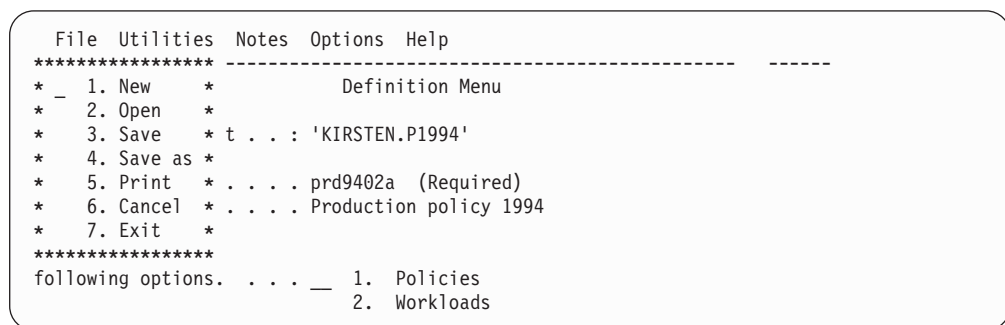


Figure 33. Definition Menu File Choices

Using the menu bar on selection lists

The menu bar is a bit different on selection lists. You type a slash next to the name of the object you want to work with in the action field. Then, move to the menu bar, select an option, and press ENTER. You then choose the desired action from the pull-down.

For example, from the service class selection list, choose the STC_1 service class, move to the menu bar on the Service-Class option, and press ENTER. Then, type 3 in the menu pull-down, and press ENTER.

```

Service-Class View Notes Options Help
*****-----
* 3 1. Create *      Service Class Selection List      ROW 14 TO 26 OF 29
* 2. Copy *
* 3. Modify *
* 4. Browse * Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
* 5. Print * Menu Bar
* 6. Delete *
* 7. Exit *
*****
Description                               Workload
--- MDLIMSR      IMS Model response          TRNIMS
--- PRDCICS
--- PRDCICSM     MSA application              PRDCICS
--- PRDIMSNR     Prod IMS non-response        PRDIMS
--- PRDIMSR      IMS response                  PRDIMS
/  STC_1        Highest Priority stc          STC
--- STC_2        High Priority stc            STC

```

Figure 34. Service Class Selection List

Using the status line

Some, but not all panels have a status line. A status line is displayed on the right side of a panel or pop-up beneath the title. The status line indicates the number of items or lines currently displayed in a list or topic, and the total size of and your current location in that list or topic.

For example, a service class selection list status line such as Row 1 to 8 of 16 states that displayed in the panel is a list of eight of the 16 service classes contained in the service definition.

Using the scrollable area

On selection list type panels, there is a scrollable area that contains a list or text. In a service class selection list you see a list of service classes. The status line shows how many service classes there are in the list. You scroll backward or forward to move through the list. Selection lists can also be pop-ups, such as selecting a workload from a list to associate it with a service class.

Using the menu bar on a selection list

On selection lists, you can select the object you want to work with before you select actions you want to perform from the menu bar.

To mark items for selection in a scrollable area, type a slash (/) over the underscore in front of the listed choice. When you select an menu bar item, the application performs the action for the marked items only.

Figure 35 on page 198 shows an example of the scrollable area on a service class selection list. The first two service classes are marked with a / for actions on the menu bar.

```

Service-Class View Notes Options Help
-----
Service Class Selection List          ROW 13 TO 24 OF 29

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Class      Description                      Workload
/      PRDIMS      IMS Production                      PRDIMS
/      PRDCICS      PRDCICS                              PRDCICS
_      PRDCICSM     MSA application                      PRDCICS

```

Figure 35. Service Class Selection List panel

Using the Action field

The action field is where you specify the action to take. On the definition menu, the action field is where you specify the workload management object that you want to work with. The action codes are standard in the application—except on a few selection lists, such as the subsystem type selection list in classification rules, and the service policy selection list. Figure 36 shows the action codes and action field on the subsystem type selection list.

Note: The menu bar pull-down choices for the file match the action codes on the selection lists. So you can choose the method according to your preference.

```

Subsystem-Type View Notes Options Help
-----
Subsystem Type Selection List for Rules  ROW 1 TO 10 OF 10

Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Type      Description                      -----Class-----
_      ASCH      IBM-defined USAA almost modified  Service Report
_      CICS      IBM-defined USAA modified         PRDCICS

```

Figure 36. Action field on the Subsystem Type Selection List panel

Using the command line

The command line is displayed according to the currently active user profile. You can issue the following commands from the command line:

=value
Repeats the previous command.

BACKWARD
Scrolls backward.

DOWN
Scrolls forward.

EXIT
Exits the panel.

FORWARD
Scrolls forward.

FKA ON|OFF

Determines whether to display the function key area.

HELP

Displays the help panel for the displayed panel.

KEYLIST

Displays the keylist utility where you can adjust PF key settings.

PFSHOW

Shows the PF key settings.

RETRIEVE

Displays any previous command.

UP Scrolls backward.

Using the function keys

The function key area at the bottom of each panel or pop-up displays actions that you can complete by pressing a function key. When a pop-up is displayed, you can press only the function keys listed in that pop-up, not the keys listed at the bottom of the panel.

Standard actions are assigned to function keys 1 through 12 and are repeated for function keys 13 through 24. The function key assignments can vary slightly, depending on options selected during installation. If you want to customize the key settings for your installation, you can use the KEYLIST utility. See “Customizing the keylists” on page 254 for more information about how to customize the key settings.

The WLM application displays function keys 1 to 12. If you want to display function keys 13 to 24, or see the standard function key settings, see *ISPF Dialog Management Guide and Reference*.

Figure 37 shows a sample of the function keys on the Service Class Selection list.

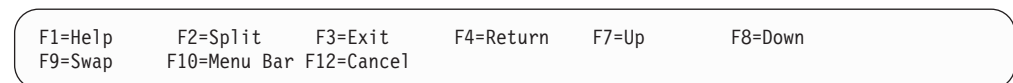


Figure 37. Function key area

Starting the WLM application

The application is shipped in the IPCS library. When you start the application, the system needs to concatenate the WLM/IPCS data sets, allocate some data sets, and then invoke the WLM panels.

To start the application, specify:

```
ex 'SYS1.SBLSCLI0(IWMARIN0)'
```

If you have different data set conventions for your IPCS data sets, or if you use storage managed data sets, you should use the WLM application exits IWMAREX1 and IWMAREX2. For more information about IWMARIN0 and how to customize the application with the WLM exits, see Appendix A, “Customizing the WLM ISPF application,” on page 247.

Now you're started

Upon entry to the interface, the WLM logo panel is displayed. Press Enter to continue, and the application displays the **Specify Definition** pop-up as shown in Figure 38. You should then specify which service definition you want to work with. You can work with one service definition at a time in the application.

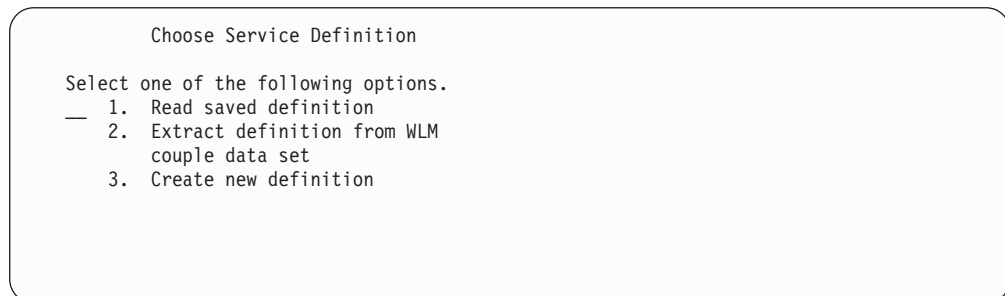


Figure 38. Choose Service Definition pop-Up

If this is your first time in the application, choose option 3, **Create a new service definition**. This option brings you to the definition menu, where you can define your service definition. Once you created a new service definition, upon exiting the application, you can either:

- Save the service definition in a PDS as ISPF tables, or in a PS as XML.
The application prompts you for a data set name, allocates the data set, and saves the service definition.
- Install the service definition on the WLM couple data set.
The install option puts the service definition currently displayed in the application out on the WLM couple data set. When installed, any changes you made to the service definition are available when you activate a policy. “Installing and extracting a service definition” on page 235 explains how to install a service definition.
- Discard the service definition.

If you want to edit a service definition previously defined and stored in an MVS data set, choose option 1, **Read saved definition**.

If you want to work with the service definition on the WLM couple data set, choose option 2, **Extract definition from WLM couple data set**. You must have previously installed a service definition on the WLM couple data set. “Installing and extracting a service definition” on page 235 explains how to extract a service definition.

Using the Definition Menu

The definition menu is the central place for entering your service definition. When you set up a service definition, you must enter a service definition name and optionally, a description on the Definition Menu.

Figure 39 on page 201 shows a sample Definition Menu with the service definition name and a description filled in.

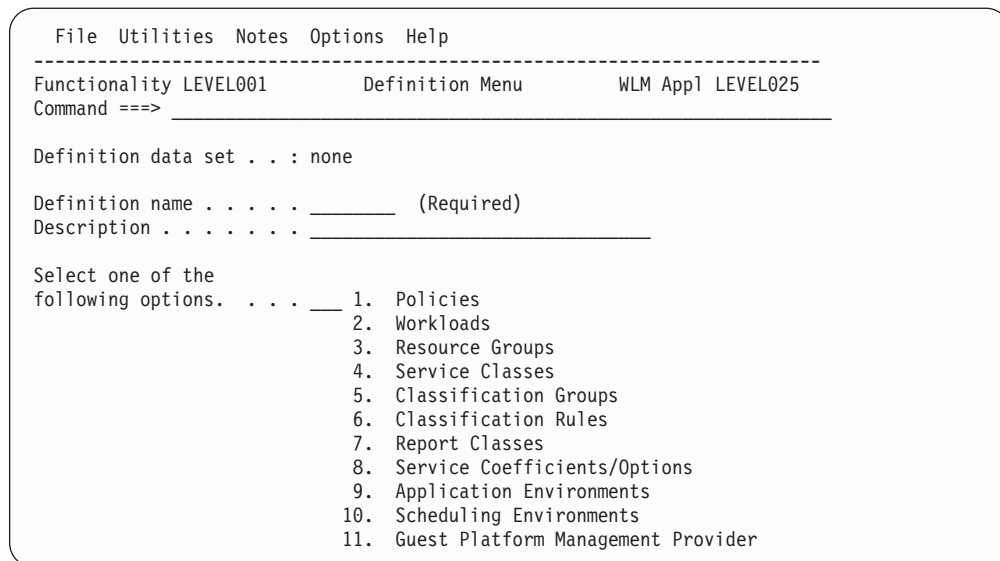


Figure 39. Definition Menu panel

When you define your service definition for the first time, you should define it in the following order:

1. Policies

A policy consists of a name, a description, and policy overrides. The first time you set up a service definition, define a policy name and description. If you do not have a business need to change your goals, you can run with one service policy, without any policy overrides.

You use a policy override only if you have a business need to change a goal for a certain time, such as for the weekend, or for nighttime. You can define your policy overrides once you have defined your service classes.

2. Workloads

A workload logically consists of a group of one or more service classes. You associate a workload with a service class in the Service Class panel. Enter your workloads before creating your service classes.

3. Resource groups (optional)

A resource group is a minimum or maximum amount of processing capacity. You associate a resource group with a service class in the Service Class panel. Enter resource groups before creating your service classes.

4. Service classes

A service class is a group of work with similar performance goals, resource requirements, or business importance. You make the association with a workload and a resource group in the service class panel. You associate a service class with incoming work in the classification rules panel. Enter service classes before creating classification rules.

Policy overrides

Once you have created a service class, you can create a policy override. You specify the policy override by selecting Service Policies from the Definition Menu, and then specifying the action code for Override service class or Override resource group.

5. Classification groups (optional)

You use groups to simplify classification. You associate a classification group with a service class in the classification rules panel. If you intend to use them,

create groups before creating classification rules. See Chapter 9, “Defining classification rules,” on page 61 for descriptions of group qualifiers.

6. Classification rules

Classification rules assign incoming work to service classes. Before you create your classification rules, you must understand which subsystem's work is represented in each of your service classes.

When you choose the option Classification Rules, you go to the Subsystem Type Selection List for Rules. This selection list is primed with all of the IBM-Supplied subsystem types. They are reserved names.

7. Report classes (optional)

A report class is a group of work for which you want reporting data. You do not have to define report classes before assigning them to work in classification rules. You can create them from within the classification rules menu.

8. Service coefficients/options

Service coefficients define the weight to be applied to one type of service over another in the calculation of service rates. You can enter new values for the CPU, IOC, MSO, and SRB service coefficients.

See “Service definition coefficients” on page 102 for more information.

There are three additional options on this panel:

- **I/O Priority Management:** The default is **no**, meaning that I/O priorities will be the same as dispatching priorities. Specifying **yes** means I/O priorities should be managed separately from dispatching priorities, according to the goals of the work. See “Specifying I/O priority management” on page 104 for more information.
- **Enable I/O Priority Groups:** The default is **no**, meaning that I/O priority groups are ignored. Specifying **yes** will cause workload management to consider I/O priority groups. Work in service classes assigned to I/O priority group HIGH always has higher I/O priority than work in service classes assigned to I/O priority group NORMAL. When you specify **yes**, you also need to specify **yes** for I/O Priority Management. See “Enabling I/O priority groups” on page 104 for more information.
- **Dynamic Alias Management:** The default is **no**, meaning that dynamic alias management is disabled for the entire sysplex. Specifying **yes** will cause workload management to dynamically reassign parallel access volume aliases to help work meet its goals and to minimize IOS queueing. See “Specifying dynamic alias management” on page 105 for more information.

9. Application Environments

An application environment is a group of application functions invoked by request and executed in server address spaces. You can have workload management start and stop these server address spaces automatically, or do this manually or through automation. You define the application environment, an optional procedure name for starting the server address spaces, and any start parameters needed for the start procedure.

10. Scheduling Environments

A scheduling environment is a list of resource names along with their required states. By associating incoming work with a scheduling environment, you ensure that work is assigned to a system only if that system satisfies all of the requirements. You define the scheduling environment, listing all of the resource names and required states that are contained within. You also define the resource names themselves.

11. Guest Platform Management Provider (GPMP)

Starts the guest platform management provider (GPMP) to allow for performance management of zEnterprise systems. For further information about the guest platform management provider, refer to Chapter 15, “Managing performance with the guest platform management provider (GPMP),” on page 147.

Using the menu bar on the Definition Menu

The menu bar on the Definition Menu has some functions not accessible from any other panel in the application. From the menu bar, you can:

- Verify a service definition
- Allocate a WLM couple data set
- Install a service definition on the WLM couple data set
- Activate a service policy

Table 17 shows the options available from the menu bar on the Definition Menu. Each of the options are explained.

Table 17. Menu bar options on the Definition Menu

File	Utilities	Notes	Options	Help
New	Install definition	Edit notepad	Process ISPF list data set	General Help
Open	Extract definition			Keys Help
Save	Activate service policy			Using Help
Save as	Allocate couple data set			Tutorial
Print	Allocate couple data set			About...
Print as GML	using CDS values			
Cancel	Validate definition			
Exit				

File

New Use new to define a new service definition.

Open Use open to read a previously defined service definition. The Read saved definition panel is displayed, where you can specify the data set name.

Save Use save to save the currently displayed service definition.

Save as

Use Save as to save the currently displayed service definition in a PDS as ISPF tables or in a PS as XML. The Save to... panel is displayed where you can specify the data set name and the save format. You do not need to preallocate the data set. If the data set does not exist, the application displays the Create Data Set? panel where you can continue with the data set create.

Print Use Print to print the complete service definition to the ISPF list data set. Use the Options menu bar option to process the ISPF list data set. This option requires no formatting step.

Print as GML

Use Print as GML for a more readable, tabular display of service definition objects and values. This option creates a source data set with GML starter set tags imbedded. The data set must be allocated as variable block with logical record length 255. This data set can then be formatted with the SCRIPT/VS processor.

Cancel

Use cancel to cancel any actions performed. Cancel is the same as using the cancel PF key.

Exit Use exit to exit from the definition menu and the application. Exit is the same as using the exit PF key.

Utilities**Install definition**

Use this option to install the service definition onto the WLM couple data set. Installing the service definition makes any changes available for policy activation.

Extract definition

Use this option to extract the service definition previously installed on the WLM couple data set.

Activate service policy

Use this option to activate a policy. When you select this option, the application displays a list of the service policies defined in the service definition currently installed on the WLM couple data set. You activate the service policy by selecting it from the list.

Note: If you have just made changes for a service definition, make sure you install it to have the changes take effect.

Allocate couple data set

Use this option to allocate both your primary and alternate WLM couple data sets. This option is for users who are allocating a WLM couple data set for the first time.

All other users should use the option "Allocate couple data set using CDS values". To make the WLM couple data set available for use in the sysplex, you must update your COUPLExx parmlib member and issue the SETXCF command.

Allocate couple data set using CDS values

Use this option to allocate both your primary and alternate WLM couple data sets based on your existing WLM couple data set size. The application displays the current size values on the panel.

To make the WLM couple data set available for use in the sysplex, you must update your COUPLExx parmlib member and issue the SETXCF command.

Validate definition

Use this option to verify that your service definition is free from certain errors that would otherwise be flagged when you attempt to save or install the service definition.

Notes**Edit notepad**

Use this option to create and edit a notepad. You can use the notepad to keep track of changes made to all parts of a service definition.

Options**Process ISPF list dataset**

Use this to process the list data set if you have previously done a Print.

Help

General Help

Use general help for information about the panel currently displayed.

Keys Help

Use keys help for information about the using PF keys.

Using Help

Use this option for information about how to get help while using the WLM application.

Tutorial

Use the tutorial option for information about how to use the panels. This options provides context-specific examples and scenarios.

About...

This option provides information about the copyright and license.

Working with service policies

When you choose the **Policy** option for the first time, the application displays the Create a Service Policy panel. Figure 40 shows a sample panel.

```
Service-Policy  Notes  Options  Help
-----
                Create a Service Policy

Enter or change the following information:

Service Policy Name . . . . . _____ (Required)
Description . . . . . _____

Command ===> _____
```

Figure 40. Create a Service Policy panel

Once you have created a service policy, any other time you choose the policy option from the definition menu, the application displays a policy selection list. From here, you can modify your policy description, print and browse your service policies, and define your service policy overrides. Figure 41 on page 206 shows a Service Policy Selection List panel.

```

Service-Policy View Notes Options Help
-----
Service Policy Selection List                                ROW 1 TO 3 OF 3

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              7=Override Service Classes, 8=Override Resource Groups,
              /=Menu Bar

Action Name      Description                                ---Last Change---
---            ---            User      Date
---            ---            ---            ---
--- HOLIDAY      Policy for shut-Down holidays KIRSTEN  1996/12/04
--- WEEKDAY      Policy for Mon - Friday      KIRSTEN  1996/12/04
--- WEEKEND      Policy for Fri - Sun         KIRSTEN  1996/12/04
***** BOTTOM OF DATA *****

Command ==> _____

```

Figure 41. Service Policy Selection List panel

Working with workloads

When you choose the **Workload** option for the first time, the application displays the Create a Workload panel. Figure 42 shows a Create a Workload panel.

```

Workload Notes Options Help
-----
Create a Workload

Enter or change the following information:

Workload Name . . . . . _____ (Required)
Description . . . . . _____

Command ==> _____

```

Figure 42. Create a Workload panel

You associate the workload with a service class in the service class panel.

Once you have created a workload, any other time you choose the workload option from the definition menu, the application displays a workload selection list. The Workload Selection List is similar to the Policy Selection List. From here, you can modify your workload description, print, and browse your workloads. Figure 43 on page 207 shows a Workload Selection List.

```

Workload View Notes Options Help
-----
Workload Selection List                                ROW 1 TO 9 OF 9

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action Name      Description                                ---Last Change---
---  ---  ---  ---  ---
---  APPC                                KIRSTEN  1996/10/07
---  BATCH  All batch                                KIRSTEN  1996/10/07
---  APIMS  IMS application                            KIRSTEN  1996/10/07
---  PRDCICS CICS Production                            ADRIAN   1996/10/17
---  PRDIMS  IMS Production                            ADRIAN   1996/10/17
---  STC    Started tasks                            KIRSTEN  1996/10/07
---  TRNCICS CICS training                            KIRSTEN  1996/10/07
---  TRNIMS  IMS training                            GEORGEN  1996/10/07
---  TSO    All TSO                                GEORGEN  1996/10/07
***** BOTTOM OF DATA *****

Command ==> _____

```

Figure 43. Workload Selection List panel

Working with resource groups

To define a resource group, choose option 3 on the Definition menu. Define a name, a description (optional), a resource group type (1, 2, or 3), a minimum and/or maximum capacity, and a memory limit. Either a capacity type or a memory limit must be specified. Associate the resource group with a service class on the service class panel. Figure 44 shows the Create a Resource Group panel. (The panels for modifying an existing resource group and overriding attributes of a resource group in a service policy look exactly the same.)

```

Resource-Group Notes Options Help
-----
Create a Resource Group

Enter or change the following information:

Resource Group Name . . . . . _____ (required)
Description . . . . . _____

Define Capacity:

--- 1. In Service Units (Sysplex Scope)
--- 2. As Percentage of the LPAR share (System Scope)
--- 3. As a Number or CPs times 100 (System Scope)

Minimum Capacity . . . . . _____
Maximum Capacity . . . . . _____

Memory Limit (System Scope):                512 GB

Command ==> _____

```

Figure 44. Create a Resource Group panel

As with a workload, once you have created a resource group, any other time you choose the resource group option from the definition menu, the application

displays a selection list. From here, you can modify your resource group description, as well as print, and browse it.

Working with service classes

Once you have defined your workloads and resource groups, you can define your service classes. Choose the **Service Class** option on the definition menu. Figure 45 shows a Create a Service Class panel. You must assign a workload to the service class in the Workload field.

```

Service-Class  Notes  Options  Help
-----
                                Create a Service Class                                ROW 1 TO 1 OF 1

Service Class Name . . . . . _____ (Required)
Description . . . . . _____
Workload Name . . . . . _____ (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO (YES or NO)
I/O Priority Group . . . . . NORMAL (NORMAL or HIGH)
Honor Priority . . . . . DEFAULT (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

      ---Period---  -----Goal-----
Action # Duration  Imp.  Description

***** BOTTOM OF DATA *****

Command ==> _____

```

Figure 45. Create a Service Class panel

Use the **Cpu Critical** field to specify CPU protection for critical regions and use the **I/O Priority Group** field to specify HIGH for I/O-sensitive work. Use the **Honor Priority** field to specify whether work in this service class is allowed to overflow to standard processors when there is insufficient specialty engine capacity for the workload demand in this service class.

Important: The use of these options limits WLM's ability to manage the system. This may affect system performance and/or reduce the system's overall throughput.

Defining goals

To enter the goal information, enter an **i** in the Action field, as shown in Figure 45. The goal selection pop-up is displayed, as shown in Figure 46 on page 209. From this panel, select the type of goal you want to assign to the service class.


```

Choose a goal type for period 1

1_ 1. Average response time
    2. Response time with percentile
    3. Execution velocity
    4. Discretionary

```

Figure 46. Choose a Goal Type pop-up

When you choose option 1, average response time, the application displays the average response time goal pop-up. There is a different pop-up for each goal type where you can fill in the information for the goal. If you are defining a single period goal, then you should not fill in a duration. If you are defining multiple periods then you must fill in a duration. Figure 47 shows an average response time goal pop-up.

```

Average response time goal

Enter a response time of up to 24 hours for period 1

Hours . . . . . _ (0-24)
Minutes . . . . . _ (0-99)
Seconds . . . . . _5_ (0-9999)

Importance . . . 2 (1=highest, 5=lowest)
Duration . . . _____ (1-999,999,999, or
                           none for last period)

```

Figure 47. Average Response Time Goal pop-up

When you press **Exit**, you return to the create a service class panel with the goal information filled in, as shown in Figure 48.

```

Service-Class Notes Options Help
-----
                          Create a Service Class                          ROW 1 TO 2 OF 2

Service Class Name . . . . . QUACK (Required)
Description . . . . . _____
Workload Name . . . . . APPC (name or ?)
Base Resource Group . . . . . _____ (name or ?)
Cpu Critical . . . . . NO (YES or NO)
I/O Priority Group . . . . . NORMAL (NORMAL or HIGH)
Honor Priority . . . . . DEFAULT (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

    ---Period--- -----Goal-----
Action # Duration Imp. Description
-----
_ 1 2 Average response time of 00:00:05.000
***** BOTTOM OF DATA *****

```

Figure 48. Create a Service Class panel

Using action codes on service class panels

You use action codes on this panel to define and edit goals. Figure 49 on page 210 shows the edit codes available for service class.

```

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

      ---Period--- -----Goal-----
Action # Duration Imp. Description
-----
  1          5    80% complete within 00:30:00.000
***** END OF DATA *****

```

Figure 49. Action Codes for Goal

I=Insert new period

Use I to define a new period. The application adds a line below. If you have multiple periods, then a duration is required on the previous period. Use action code E to edit the previous period. You'll go through the windows with the goals information filled in, and you can add a duration.

E=Edit period

Use E to edit a period.

D=Delete period

Use D to delete a period. If you have defined multiple periods for a service class, remember that you do not define a duration for the last period.

Defining service policy overrides

You define all service policy overrides from the service policy selection list. You can define three kinds of service policy overrides:

- Override service class goals
- Override resource group assignment
- Override resource group attributes.

To override a service class goal, choose either the action code or the menu bar option to Override Service Classes. Figure 50 shows a service policy selection list where you have chose to override the service classes for the weekend policy.

```

Service-Policy View Notes Options Help
-----
Service Policy Selection List          ROW 1 TO 3 OF 3

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              7=Override Service Classes, 8=Override Resource Groups,
              /=Menu Bar

Action Name      Description                               ---Last Change---
-----
  1  HOLIDAY     Policy for shut-Down holidays          KIRSTEN  1996/12/04
  2  WEEKDAY     Policy for Mon - Friday                KIRSTEN  1996/12/04
  7  WEEKEND     Policy for Fri - Sun                   KIRSTEN  1996/12/04
***** BOTTOM OF DATA *****

Command ==> _____

```

Figure 50. Service Policy Selection List panel

The application displays the Override Service Class Selection List panel. This is the list of all of your defined service classes, similar to the Service Class Selection list, except it has an extra field called **Overridden Goal** which indicates whether or not

the service class goal or resource group assignment was overridden for that policy. Figure 51 shows an Override Service Class Selection List. Select a service class whose goal or resource group assignment you want to change and specify the action code or menu bar option to override the service class.

```

Override View Notes Options Help
-----
IWMAP4A          Override Service Class Selection List          ROW 1 TO 8 OF 14

Service Policy Name . . . . : BACKUPS

Action Codes: 3=Override Service Class, 4=Browse, 5=Print,
              6=Restore Base attributes, /=Menu Bar

Action  Service  Overridden
Class  Class  Goal      Description
---    -
   BATCHJ  NO       Class J work
   BATCHTST NO       Test
   BATCHX  NO       Class X work
   CICSFAST NO      Fast CICS work
   CICSLOW NO      Slow CICS work
3_  HOTBATCH NO      Hot batch
   IMSNRESP NO     IMS non response
   IMSRESP  NO     IMS response

Command ==>>> _____

```

Figure 51. Override Service Class Selection List panel

When you choose the override service class option, the application displays the Override Attributes for a Service Class panel. Figure 52 on page 212 shows an Override Attributes for a Service Class panel. To override the goal, use the same codes to edit the goal as you do on the Create or Modify a Service Class panel. You can also change the **Cpu Critical**, **I/O Priority Group**, and **Honor Priority** settings, just as you do on the Create or Modify a Service Class panel. To change the resource group assignment of the service class, either enter in a resource group name, or put a ? in the resource group field to select a resource group from the selection list.

If you want to remove a service from a resource group, blank out the name.

```

Service-Class Xref Notes Options Help
-----
Override attributes for a Service Class      ROW 1 TO 2 OF 2

Service Policy Name . . . . . : STANDARD
Service Class Name . . . . . : BAT_0

Override the following information:
Resource Group . . . . . : _____ (name or ?)
Cpu Critical . . . . . : NO (YES or NO)
I/O Priority Group . . . . . : NORMAL (NORMAL or HIGH)
Honor Priority . . . . . : DEFAULT (DEFAULT or NO)

Action Codes: I=Insert new period, E=Edit period, D=Delete period.

      ---Period--- -----Goal-----
Action # Duration Imp. Description
-----
  1      1          2 Execution velocity of 70
***** BOTTOM OF DATA *****

Command ==> _____

```

Figure 52. Override Attributes for a Service Class panel

Once you have edited your goal or changed the resource group assignment, press **Exit**, and you return to the Override Service Class Selection List. The **Overridden Goal** field for that service class now says YES.

Overriding resource group attributes works the same way. However, when you override the memory limit of a resource group with a lower value, the new lower memory limit is ignored when you activate the policy.

Working with classification rules

Classification of work depends on having the rules that are defined for the correct subsystem type. When you choose the **Classification Rules** option from the Definition menu, you go to the Subsystem Type Selection List for Rules panel. This panel initially contains the reserved names of the IBM-supplied subsystem types.

Although you might want to change the description of the subsystem types, do not delete any of the entries that are provided by IBM unless your installation does not plan to ever use them. If your installation later does need to use them, they can be manually added then.

Figure 53 on page 213 shows the subsystem type selection panel:

```

Subsystem-Type View Notes Options Help
-----
Subsystem Type Selection List for Rules      ROW 1 TO 15 OF 15

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Type      Description
-----
_      ASCH      Use Modify to enter YOUR rules
_      CB        Use Modify to enter YOUR rules
_      CICS      CICS interactive
_      DB2      Use Modify to enter YOUR rules
_      DDF      Use Modify to enter YOUR rules
_      EWLM     Use Modify to enter YOUR rules
_3     IMS      Use Modify to enter YOUR rules
_      IWEB     Use Modify to enter YOUR rules
_      JES      batch
_      LSFM     Use Modify to enter YOUR rules
_      MQ        Use Modify to enter YOUR rules
_      OMVS     Use Modify to enter YOUR rules
_      SOM      Use Modify to enter YOUR rules
_      STC      Use Modify to enter YOUR rules
_      TSO      TSO

-----Class-----
Service  Report

***** End of data *****

Command ==> _____

```

Figure 53. Subsystem Type Selection List for Rules panel

To create your rules, use the **Modify** option (3) to create rules for the IBM-supplied subsystem types. For example, use the modify option on subsystem type IMS as show in Figure 53 to create the rules for your IMS work. Figure 54 shows the Modify Rules for the Subsystem Type panel.

```

Subsystem-Type Xref Notes Options Help
-----
Modify Rules for the Subsystem Type      Row 1 to 2 of
Command ==> _____                  SCROLL ==> PAG

Subsystem Type . : CICS      Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row R=Repeat   IS=Insert Sub-rule
                                   More==>

Action  -----Qualifier-----
Type   Name      Start
-----
_      1  SI      IMSTRN*  _
_      2  TC      15      _
_      1  SI      IMSMDL*  _
_      2  TC      15      _
_      1  TC      15      _

-----Class-----
Service  Report
DEFAULTS: PRDIMSR  _____
          TRNIMSR  _____
          TRNIMSNR _____
          MDLIMSR  _____
          MDLIMSNR _____
          PRDIMSNR _____

Command ==> _____

```

Figure 54. Modify Rules for the Subsystem Type panel

The **Fold qualifier names** option, set to the default Y, means that the qualifier names are folded to uppercase as soon as you type them in and then press Enter. If you set this option to N, then the qualifier names remain in the case they are typed in. Leave this option set to Y unless you know that you need mixed case qualifier names in your classification rules.

While you are creating rules, you can scroll right (PF11) to complete the description fields. Figure 55 shows the *Modify Rules for the Subsystem Type* panel, scrolled right to the description fields.

```

Subsystem-Type Xref Notes Options Help
-----
                          Modify Rules for the Subsystem Type          Row 1 to 2 of
Command ==>> _____ SCROLL ==>> PAG

Subsystem Type . : CICS          Fold qualifier names?  Y (Y or N)
Description . . . IBM-defined subsystem type

Action codes:  A=After   C=Copy   M=Move   I=Insert rule
                B=Before  D=Delete row R=Repeat IS=Insert Sub-rule
                <=== More ===>

-----Qualifier-----      -----Description-----
Action  Type   Name   Start
-----  ---   ----   -----
_____ 1  SI    IMSTRN*  _____ (Descriptions here..._____
_____ 2  TC    15      _____ Any valid characters,_____
_____ 1  SI    IMSDL*   _____ up to 32 in length.)_____
_____ 2  TC    15      _____
_____ 1  TC    15      _____

Command ==>> _____

```

Figure 55. *Modify Rules for the Subsystem Type* panel, scrolled right to description fields

After completing a description, you can scroll right (PF11) yet again to complete the **Storage Critical** and **Reporting Attribute** fields. For JES and STC work only, you can also complete the **Manage Regions Using Goals Of** field, with one of these values:

- TRANSACTION. The region is managed to the transaction response time goals.
- REGION. The region is managed to the goal of the service class that is assigned to the region.
- BOTH. The region is managed to the goal of the service class that is assigned to the region, but nevertheless tracks all transaction completions so that WLM can still manage the transaction service classes according to their response time goals.

For all subsystem types except JES and STC work, the **Manage Regions Using Goals Of** field contains N/A. See Chapter 12, “Defining special protection options for critical work,” on page 109 for more information on using these fields.

Important: The use of the Storage Critical and Manage Region Using Goals Of options limits WLM's ability to manage the system. This may affect system performance and reduce the system's overall throughput.

The Reporting Attribute option lets you specify which transactions are mobile transactions that are eligible for mobile workload pricing. WLM gathers and accumulates CPU service separately for each value of the Reporting Attribute, and reports it at the service and report class level, and at the system level. See “Defining special reporting options for workload reporting” on page 82 for more information on using this field.

Figure 56 on page 215 shows the *Modify Rules for the Subsystem Type* panel, scrolled right to the **Storage Critical**, **Reporting Attribute**, and **Manage Regions Using Goals Of** fields.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 2 of
SCROLL ==> PAG

Subsystem Type . : STC Fold qualifier names? Y (Y or N)
Description . . . IBM-defined subsystem type

Action codes: A=After C=Copy M=Move I=Insert rule
               B=Before D=Delete row R=Repeat IS=Insert Sub-rule
               <=== More

-----Qualifier-----
Action Type Name Start Storage Reporting Manage Region
-----
1 TN COMBL* NO NONE TRANSACTION
2 UI COMBLD NO NONE TRANSACTION
2 UI COMFTP YES MOBILE TRANSACTION
1 UI SRVLIB YES NONE TRANSACTION
1 PF 99 YES NONE REGION
1 TC B NO NONE BOTH
1 TC C NO NONE REGION

```

Figure 56. Modify Rules for the Subsystem Type panel, scrolled right to Storage Critical, Reporting Attribute, and Manage Regions Using Goals Of fields

Keep your classification rules as simple as possible. Use subsystem defaults to cover most cases, and list the exceptions. Remember the rules are order sensitive, and the first matched rule applies.

If you have a non-IBM subsystem type, you can use Create to create your subsystem type. You can also modify the qualifier selection list to include only the qualifier types that apply to your subsystem type.

For more information about defining the order of rules, and how to use the **Start** field, see Chapter 9, “Defining classification rules,” on page 61.

Using action codes on the Modify Rules panel

You use action codes on this panel to create and change your classification rules.

```

Subsystem-Type Xref Notes Options Help
-----
Command ==> _____ Modify Rules for the Subsystem Type Row 1 to 2 of 2
SCROLL ==> PAGE

Subsystem Type . : CICS Fold qualifier names? Y (Y or N)
Description . . . IBM-defined subsystem type

Action codes: A=After C=Copy M=Move I=Insert rule
               B=Before D=Delete row R=Repeat IS=Insert Sub-rule

```

Figure 57. Action codes for classification rules

In the **Qualifier Type** field, enter the type of qualifier that you are defining. A qualifier type can be one of the following:

- AI** Accounting information
- AIG** Accounting information group
- CAI** Client accounting information
- CAIG** Client accounting information group
- CI** Correlation information
- CIG** Correlation information group

CIP Client IP address
CIPG Client IP address group
CT Connection type
CTG Connection type group
CTN Client transaction name
CTNG
 Client transaction name group
CUI Client userid
CUIG Client userid group
CWN Client workstation name
CWNG
 Client workstation name group
ESC zEnterprise service class name from a Unified Resource Manager
 performance policy
LU LU name
LUG LU name group
NET Net ID
NETG Net ID group
PC Process name
PCG Process name group
PF Perform
PFG Perform group
PK Package name
PKG Package name group
PN Plan name
PNG Plan name group
PR Procedure name
PRG Procedure name group
PRI Priority
PX Sysplex name
PXG Sysplex name group
SE Scheduling environment name
SEG Scheduling environment name group
SI Subsystem instance
SIG Subsystem instance group
SPM Subsystem parameter
SPMG
 Subsystem parameter group
SSC Subsystem collection name

SSCG Subsystem collection name group
SY System name
SYG System name group
TC Transaction class
TCG Transaction class group
TN Transaction name
TNG Transaction name group
UI Userid
UIG Userid group

You can either enter the qualifier type, or select it from a selection list. To select it from a list, enter a **?** in the **Qualifier Type** field. "Using selection lists for classification rules" on page 218 explains more about using selection lists.

Then type in the name of the qualifier in the **Name** field, and the service class in the **Service Class** field.

You can use the action codes on this panel to enter in additional classification rules. The action codes are:

I=Insert rule

To create a rule at the same level as the rule you type the action code next to.

IS=Insert Sub-rule

To create a next level rule or a nest. IS=Insert Sub-rule specifies a rule under the previous rule; that is, a rule at the next indented level.

R=Repeat

To copy a rule and its sub-rules, if any. The copied rule is placed directly below the one you are copying. If a rule has sub-rules; that is, it is a family, the entire family is copied.

D>Delete row

To delete a rule.

A=After

Use with C=Copy and M=Move to specify that the copied/moved rule is to go after this rule.

B=Before

Use with C=Copy and M=Move to specify that the copied/moved rule is to go before this rule.

C=Copy

To copy a rule and its sub-rules, if any. Use B=Before or A=After with C=Copy to specify where to copy the rule. The rule is copied at the same level as the rule you are placing it Before or After. For example, a level 3 rule becomes a level 2 rule if you copy it before or after a level 2 rule.

When a rule changes levels as a result of a Copy, the levels of any sub-rules that were copied with it are bumped up or down accordingly. The rule family is copied at the same level as the Before or After rule.

M=Move

To move a rule and its sub-rules, if any. Use B=Before or A=After with

M=Move to specify where to move the rule. The rule is moved according to the way you do a copy, by placing it Before or After.

Using selection lists for classification rules

For a list of applicable qualifiers for the subsystem type, enter ? in the qualifier type field. You can select a qualifier from the list by entering / next to the one you want to use.

For a list of the defined service classes enter ? in the service class field. You can select a service class from the list by entering / next to the one you want to use.

You can define a report class by typing the report class name in the report class field. A pop-up panel lets you confirm the create. For a list of the defined report classes, enter ? in the report class field.

Creating a subsystem type for rules

Use the **Create** option only if your installation has its own subsystem type, or a vendor subsystem type that supports workload management. You must check that product's documentation for its reserved subsystem type name.

Deleting a subsystem type for rules

Use the **Delete** option on the Subsystem Type Selection List for Rules panel only to remove an IBM-supplied subsystem type that your installation does not have, or does not plan on using. The application displays a pop-up for confirmation of the delete.

Working with classification groups

If you have a long list of work that you want to use in a classification rule, you can create a group. You can create groups for all qualifier types except for Priority and zEnterprise Service Class.

For example, you may want to create a transaction name group to use in classification rules for your started tasks. Figure 58 on page 219 shows a Create a Group panel. You can use wildcard and masking notation in qualifier groups. For work qualifiers that run longer than 8 characters, you can use a start position to indicate how far to index into the string. Note that there is also room next to each classification group for a description.

```

Group  Xref  Notes  Options  Help
-----
                                Create a Group                                ROW 1 TO 10 OF 17

Enter or change the following information:

Qualifier type . . . . . : Transaction Name
Group name . . . . . : STC_GR1
Description . . . . . : Low Priority STC
Fold qualifier names? . . . . Y (Y or N)

Qualifier Name  Start  Description
UCC7           ___    (Descriptions here...___
WSF2*          ___    Any valid characters,___
PHOENIX        ___    up to 32 in length.)___
NVDM           ___    _____
BMC*           ___    _____
DBUS*          ___    _____
DFHSM*         ___    _____
EMAIL          ___    _____
NETEX          ___    _____

```

Figure 58. Create a Group panel

Then, you go to modify the STC subsystem on the Subsystem Type Selection List for Rules panel, and reference the group in the rules. Choose transaction name group for the qualifier type, and enter the name of the group. You can also enter ? in the qualifier name field for a list of the defined groups.

Figure 59 shows a Modify Rules for STC Subsystem panel. On this panel, you specify TNG in the qualifier type field. Then you can enter the transaction group name STC_GR1 in the **Qualifier Name** field.

```

Subsystem-Type Xref  Notes  Options  Help
-----
                                Modify Rules for the Subsystem Type                                Row 1 to 7 of 8

Subsystem Type . . . . . : STC
Description . . . . . : IBM-defined subsystem type
Fold qualifier names? . . . . Y (Y or N)

Enter one or more action codes: A=After B=Before C=Copy D=Delete
M=Move I=Insert rule IS=Insert Sub-rule R=Repeat

Action  -----Qualifier-----          -----Class-----
Type   Name      Start          Service      Report
_____ 1 TNG      STC_GR1      ___          DEFAULTS: STC_5
                                     STC_1

```

Figure 59. Modify Rules for STC Subsystem

Working with report classes

You can create a report class by either:

- Selecting the **Report Classes** option on the Definition Menu
- Typing in a name on the report class field of the Modify Rules for the Subsystem Type panel.

When you select the report classes option on the Definition menu, you go to the Create a Report Class panel. From there, you enter the report class name and

description. The next time you choose the option from the Definition Menu, the application displays Report Class selection list.

To create a report class from the Modify Rules for Subsystem panel, enter a name in the report class field. Figure 60 shows a Modify Rules for the CICS Subsystem panel. If you type in a name, for example CICSHOT, in the report class field, the application displays a Create Report Class pop-up panel. Figure 61 shows a Create a Report Class pop-up.

```

Subsystem-Type Xref Notes Options Help
Modify Rules for the Subsystem Type Row 1 to 6 of
Command ==> _____ SCROLL ==> PAG

Subsystem Type . : CICS Fold qualifier names? Y (Y or N)
Description . . . IBM-defined USAA modified

Action codes: A=After C=Copy M=Move I=Insert rule
               B=Before D=Delete row R=Repeat IS=Insert Sub-rule
-----Qualifier-----
Action Type Name Start Service Report
-----Class-----
_____ 1 TN MSA* _____ PRDCICS CICSHOT_
_____ 1 SI CICSTRN* _____ PRDCICSM _____
TRNCICS _____
***** BOTTOM OF DATA *****

```

Figure 60. Creating a report class on the Modify Rules panel

```

Create a Report Class

An undefined report class was specified.

Press ENTER to define the class, or cancel to quit.

Report Class name : CICSHOT
Description . . . . _____

```

Figure 61. Create a Report Class confirmation panel

When you press Enter, the report class is created, and is added to the report class selection list. You can type ? in the report class field on the Modify Rules for a Subsystem Type panel for a selection list of report classes.

Working with service coefficients and options

On the Service Coefficient/Service Definition Option panel, you can specify the service definition coefficients, whether you want workload management to manage your I/O priorities, and whether you want workload management to manage your parallel access volume alias addresses.

Chapter 11, “Defining service coefficients and options,” on page 101 provides some advice on how and when to adjust your coefficients.

To find out what service definition coefficients you are currently running with, check your RMF Monitor I Workload Activity Report. This report lists the service definition coefficients your installation is currently using.

If you want workload management to manage I/O priorities for you, specify YES for **I/O priority management**. When you specify YES, workload management manages your I/O priorities and includes I/O usings and delays in its execution velocity calculation. When you specify NO (which is the default), workload

management sets the I/O priority to be the same as the dispatching priority, and I/O usings and delays are not included in the execution velocity calculation.

If you want workload management to work with I/O priority groups, specify YES to ensure that work managed by a service class assigned to I/O priority group HIGH always has a higher I/O priority than work in group NORMAL. This protection can be valuable for work which is extremely I/O-sensitive. See “Enabling I/O priority groups” on page 104 for more information on this setting.

If you want workload management to manage parallel access volume alias addresses for you, specify YES for **Dynamic alias management**. When you specify YES, workload management dynamically reassigns alias addresses from one base to another to help work meet its goals and to minimize IOS queueing. See “Specifying dynamic alias management” on page 105 for more information on this global setting and its relationship to the **WLMPAV=** setting on each individual device. When you specify NO, which is the default, dynamic alias management is globally disabled for the entire sysplex. Systems will still use the aliases assigned to the base devices, but there will be no automatic reassignment of aliases based on goals or queueing.

Figure 62 shows the Service Coefficient/Service Definition Options panel.

```

Coefficients/Options  Notes  Options  Help
-----
                Service Coefficient/Service Definition Options

Enter or change the Service Coefficients:

CPU  . . . . . _____ (0.1-99.9)
IOC  . . . . . _____ (0.0-99.9)
MSO  . . . . . _____ (0.0000-99.9999)
SRB  . . . . . _____ (0.0-99.9)

Enter or change the service definition options:

I/O priority management . . . . . NO (Yes or No)
Enable I/O priority groups . . . . . NO (Yes or No)
Dynamic alias management . . . . . NO (Yes or No)

Command ==> _____

```

Figure 62. Service Coefficients panel

Working with application environments

Define an application environment so that workload management can assist in managing work that runs in server address spaces. You can specify that workload management should dynamically start and stop server address spaces to process work running in the application environment. You can define application environments for the DB2, SOM, and IWEB IBM-supplied subsystem types. Refer to subsystem reference information for guidance on how to specify application environments for the subsystem.

Figure 63 on page 222 shows the Application Environments panel.

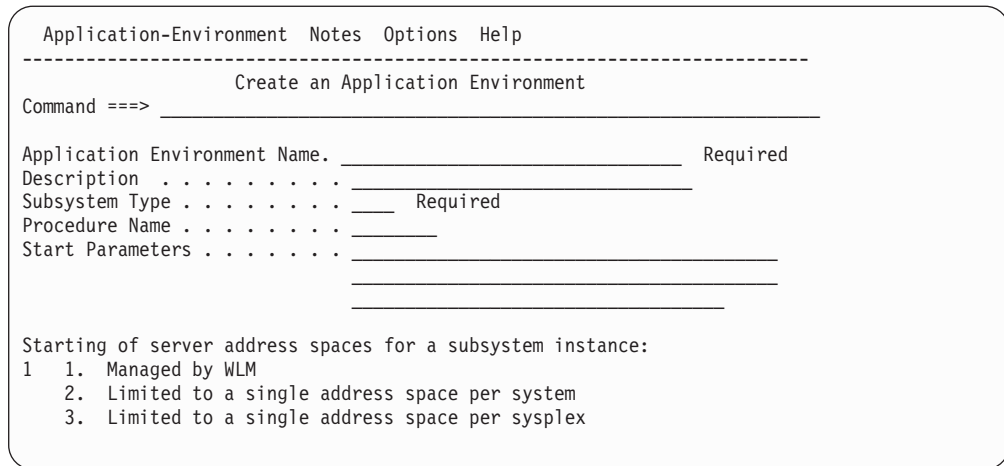


Figure 63. Create an Application Environments panel

Working with scheduling environments

Use the scheduling environment panels to define scheduling environments and their lists of resource names and required states, and to define the individual resource names themselves.

Creating a new scheduling environment

If you have not yet defined any scheduling environments, Figure 64 shows the first panel you will see.

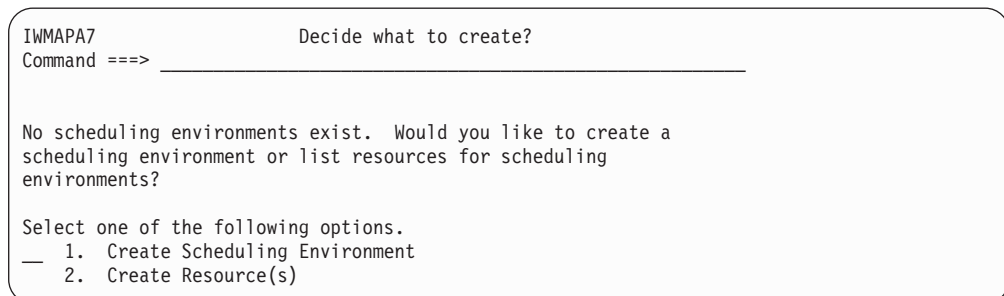


Figure 64. Decide What to Create panel

From here you can either enter **1** to create a scheduling environment, or a **2** to create resources. (You might want to create all of your resources, for instance, before creating your first scheduling environment. This is covered in “Creating a new resource directly from the main panel” on page 227.)

Note: Until you have created at least one scheduling environment (no matter how many resources you create), you will always see this initial panel.

Figure 65 on page 223 shows the main Scheduling Environment Selection List panel. Assuming that you have at least one scheduling environment already created, this is the first panel you see when you work with scheduling environments.

```

Scheduling-Environments  Notes  Options  Resources  Help
-----
IWMAPAA                Scheduling Environment Selection List          Row 1 to 2 of 2
Command ==>>> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Scheduling Environment Name      Description
 1_     IMSPRIME                          Prime IMS Processing
  _     CICS_SPEC                          Special CICS Processing

```

Figure 65. Scheduling Environment Selection List panel

By entering **1** on any Action line, you will go to the Create a Scheduling Environment panel, as shown in Figure 66. (You can also get there by putting the cursor under the Scheduling-Environment field at the top of the screen, and then entering **1** in the pop-up box.)

```

Scheduling-Environments  Notes  Options  Help
-----
IWMAPAD                Create a Scheduling Environment          Row 1 to 1 of 1
Command ==>>> _____

Scheduling Environment Name  DB2LATE_____ Required
Description . . . . . Offshift DB2 Processing_____

Action Codes: A=Add  D=Delete

Action  Resource Name      Required
 a_     _____          _____

```

Figure 66. Create a Scheduling Environment panel

Enter the scheduling environment name and description, and then enter **A** on the Action line to go to the next panel, Resource Definition List, as seen in Figure 67.

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAS                Resource Definition List          Row 1 to 4 of 4
Command ==>>> _____

Selection For Scheduling Environment DB2LATE

Action Codes: A=Add  S=Select  X=XREF  /=Menu Bar

Action  Resource Name      In Use  Resource Description
  _     CICS001            YES     CICS Subsystem
 S_     DB2A              YES     DB2 Subsystem
  _     IMS1             YES     IMS Subsystem
 S_     PRIMETIME         YES     Peak Business Hours

```

Figure 67. Resource Definition List panel

This panel shows *all* of the resource definitions in the service definition. The YES in the **In Use** field simply means that the resource name is currently a part of some scheduling environment. (It does *not* mean that it is in use by the scheduling environment you are defining when you get to this screen, in this case DB2LATE.)

You can select as many resources as you like. In this case, we'll select DB2A and PRIMETIME. Both have already been defined (although only PRIMETIME is currently being used by another scheduling environment). To make the resource name part of the scheduling environment, enter **S** on the Action line next to it.

(Alternately, you can enter / next to the resource names and then put the cursor under the **Resources** field at the top of the screen and press Enter. Then select 2 from the options in the pop-up box.)

If you have many resources, you can use the **LOCATE** primary command to scroll the display to a particular resource name.

Note: You can create new resources from this panel, as well. See “Creating a new resource while working with a scheduling environment” on page 228.

This will take you back to the Create a Scheduling Environment panel, as shown in Figure 68.

```

Scheduling-Environments  Notes  Options  Help
-----
IWMAPAD                  Create a Scheduling Environment          Row 1 to 2 of 2
Command ==>>> _____

Scheduling Environment Name  DB2LATE_____ Required
Description . . . . . Offshift DB2 Processing_____

Action Codes: A=Add  D=Delete

Action  Resource Name      Required
-----  -
  _____  DB2A                ON_____  DB2 Subsystem
  _____  PRIMETIME           OFF_____  Peak Business Hours

```

Figure 68. Create a Scheduling Environment panel

The DB2A and PRIMETIME names and descriptions are automatically brought back to this panel. Now you must specify either ON or OFF under the **Required State** field. In this example, we have chosen ON for DB2A and OFF for PRIMETIME (meaning that the DB2A resource state must be set to ON and the PRIMETIME resource state must be set to OFF for this scheduling environment to be satisfied).

Modifying a scheduling environment

Once a scheduling environment is created, you can modify it at any time, either changing the required states of resources, adding new resources to the scheduling environment, or deleting resources from the scheduling environment.

Now that DB2LATE has been created in the previous section, here's how we would modify it. From the main Scheduling Environment Selection List panel, enter a 3 in the Action line next to the scheduling environment you want to change, as shown in Figure 69 on page 225. (Alternately, you can enter a / next to it, put your cursor under the Scheduling-Environments field at the top of the screen, and then select 3 from the pop-up box.)

If you have many scheduling environments, you can use the LOCATE primary command to scroll the display to a particular name.


```

Scheduling-Environments  Notes  Options  Resources  Help
-----
IWMAPAJ                Scheduling Environment Selection List          Row 1 to 3 of 3
Command ==>>> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Scheduling Environment Name      Description
3_     DB2LATE                             Offshift DB2 Processing
_      IMSPRIME                             Prime IMS Processing
_      CICS_SPEC                            Special CICS Processing

```

Figure 69. Scheduling Environment Selection List panel

This will take you to the Modify a Scheduling Environment panel. Here, you can add and delete resources, and change required states. In Figure 70, we could delete DB2A, and change the required state for PRIMETIME from OFF to ON. (To add new resources from this panel, see “Creating a new resource while working with a scheduling environment” on page 228.) When you press enter, you'll get a confirmation message, telling you to press EXIT to save your changes, or CANCEL to discard them.

```

Scheduling-Environments  Notes  Options  Help
-----
IWMAPAJ                Modify a Scheduling Environment          Row 1 to 2 of 2
Command ==>>> _____

Scheduling Environment Name  DB2LATE_____ Required
Description . . . . . Offshift DB2 Processing_____

Action Codes: A=Add D=Delete

Action  Resource Name      Required State      Resource Description
d_     DB2A                ON_____      DB2 Subsystem
_      PRIMETIME          ON_____      Peak Business Hours

```

Figure 70. Modify a Scheduling Environment panel

Important: When you delete a resource name from this panel, you are simply deleting it *from the scheduling environment*. The resource name will still exist on the Resource Definitions List panel. To delete the resource definition itself, see “Deleting a resource” on page 231.

Copying a scheduling environment

To copy a scheduling environment, from the main Scheduling Environment Selection List panel (as shown in Figure 69) enter a **2** on the Action line next to the name. This will bring you to the Copy a Scheduling Environment panel, as shown in Figure 71 on page 226.

```

IWMAPAC                               Copy A Scheduling Environment           Row 1 to 2 of 2
Command ==>>

Scheduling Environment Name  NEWSCHENV_____ Required
Description . . . . . _____

Action Codes: A=Add  D=Delete

Action  Resource Name      Required
-----  -----
  _     DB2A                ON       DB2 Subsystem
  _     PRIMETIME          OFF      Peak Business Hours

```

Figure 71. Copy a Scheduling Environment panel

You must give the new scheduling environment a new, unique name. From this screen, you can modify the new scheduling environment, as shown in “Modifying a scheduling environment” on page 224.

The copy function is useful when you are working with scheduling environments with large numbers of resource names. If you wish to create a new scheduling environment that is similar to an existing one, you can simply copy the original and then make whatever changes are necessary.

Browsing a scheduling environment

To browse a scheduling environment, from the main Scheduling Environment Selection List panel (as shown in Figure 69 on page 225) enter a 4 on the Action line next to the name. You'll see a browse screen as shown in Figure 72.

```

IWMAPUB                               Line 00000000 Col 001 072
Command ==>>                               SCROLL ==>> PAGE
***** Top of Data *****

Scheduling Environment Name. . DB2LATE
Description. . . . . Offshift DB2 Processing

Resource Name      State  Resource Description
-----
DB2A                ON    DB2 Subsystem
PRIMETIME          OFF   Peak Business Hours
***** End of Data *****

```

Figure 72. Browse a Scheduling Environment panel

Printing a scheduling environment

To print a scheduling environment, from the main Scheduling Environment Selection List panel (as shown in Figure 69 on page 225) enter a 5 on the Action line next to the name. The output will be written to your ISPF list data set.

Deleting a scheduling environment

To delete a scheduling environment, from the main Scheduling Environment Selection List panel (as shown in Figure 69 on page 225) enter a 6 on the Action line next to the name. This will bring you to a confirmation screen, as shown in Figure 73 on page 227.

```

-----
IWMAPAX          Delete A Scheduling Environment
Command ===> _____

Scheduling Environment Name : DB2LATE
Description . . . . . : Offshift DB2 Processing
-----

Confirm the deletion request for the above scheduling environment.

Response . . . . . yes (Yes or No)

```

Figure 73. Delete a Scheduling Environment panel

Note that all of the resource definitions that were part of this scheduling environment *still exist*. However, they are no longer members of this particular scheduling environment anymore.

Creating a new resource

In the example shown in Figure 67 on page 223, as we were creating a new scheduling environment (DB2LATE), we chose resource names (DB2A and PRIMETIME) that already existed. What if we had needed to add a new resource name called VECTOR? As you saw on that Resource Definition List panel, VECTOR did not yet exist.

There are two ways to add a new resource definition:

- Directly from the main panel
- While working with a scheduling environment

Creating a new resource directly from the main panel

From the main Scheduling Environment Selection List panel, put the cursor under the Resources field at the top of the screen and press Enter. When the pop-up box appears, enter a 1, as shown in Figure 74. (In this pop-up, 1 is the only option.)

```

Scheduling-Environments  Notes  Options  Resources  Help
-----
Scheduling Environment * 1 1. Process Resources * to 3 of 3
Command ===> _____ *****

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Scheduling Environment Name  Description
-----
_      CICS_SPEC                      Special CICS Processing
_      DB2LATE                       Offshift DB2 Processing
_      IMSPRIME                     Prime IMS Processing

```

Figure 74. Scheduling Environment Selection List panel

This will take you directly to the Resource Definition List panel, where you can add new resources. Enter an A on any Action line, as shown in Figure 75 on page 228. (Alternately, you can put a / next to the resource names and then put the cursor under the Resources field at the top of the screen and hit ENTER. Then select 1 from the options in the pop-up box.)

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAR          Resource Definition List          Row 1 to 4 of 4
Command ==>>> _____

Action Codes: A=Add  D=Delete  X=XREF  /=Menu Bar

Action  Resource Name  In Use  Resource Description
--      CICS001          YES     CICS Subsystem
--      DB2A             YES     DB2 Subsystem
--      IMS1           YES     IMS Subsystem
a_     PRIMETIME        YES     Peak Business Hours

```

Figure 75. Resource Definition List panel

This would take you to the Define Resource panel, where you would enter the information for the new resource name, as shown in Figure 76.

```

IWMAPAE          Define Resource
Command ==>>> _____

Resource name . . . . . VECTOR_____ Required
Resource Description . . . . . Vector Processor_____

```

Figure 76. Define Resource panel

When we return to the Resource Definition List panel, the new resource name is there, as shown in Figure 77.

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAR          Resource Definition List          Row 1 to 5 of 5
Command ==>>> _____

Action Codes: A=Add  D=Delete  X=XREF  /=Menu Bar

Action  Resource Name  In Use  Resource Description
--      CICS001          YES     CICS Subsystem
--      DB2A             YES     DB2 Subsystem
--      IMS1           YES     IMS Subsystem
--      PRIMETIME        YES     Peak Business Hours
--      VECTOR          YES     Vector Processor

```

Figure 77. Resource Definition List panel

Note that the panel automatically alphabetizes the resource names. Note also that VECTOR is not yet shown as In Use. We would need to go create or modify a scheduling environment and then select VECTOR.

Creating a new resource while working with a scheduling environment

While you are in the middle of creating or modifying a scheduling environment, you can take this shortcut method to create new resources. In this case where you wanted to create the new VECTOR resource and add it to DB2LATE, this method would be the logical choice.

From either the Create a Scheduling Environment panel or the Modify a Scheduling Environment panel (they look very similar), enter an A on any Action line, as shown in Figure 78 on page 229.

```

Scheduling-Environments  Notes  Options  Help
-----
IWMAPAJ                Modify a Scheduling Environment          Row 1 to 2 of 2
Command ==>>> _____

Scheduling Environment Name  DB2LATE_____ Required
Description . . . . . Offshift DB2 Processing_____

Action Codes: A=Add  D=Delete

Action  Resource Name      Required
-----  -
a_     PRIMETIME             OFF_____ Peak Business Hours

```

Figure 78. Modify a Scheduling Environment panel

This would take you to the Resource Definition List panel, where you can enter an A on any Action line, as shown in Figure 79. (Alternately, you can enter a / next to the resource names and then put the cursor under the **Resources** field at the top of the screen and press Enter. Then select 1 from the options in the pop-up box.)

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAS                Resource Definition List          Row 1 to 4 of 4
Command ==>>> _____

Selection For Scheduling Environment DB2LATE

Action Codes: A=Add  S=Select  X=XREF  /=Menu Bar

Action  Resource Name      In Use  Resource Description
-----  -
_       CICS001            YES     CICS Subsystem
_       DB2A                YES     DB2 Subsystem
_       IMS1                YES     IMS Subsystem
a_     PRIMETIME           YES     Peak Business Hours

```

Figure 79. Resource Definition List panel

As in the method described in “Creating a new resource directly from the main panel” on page 227, from here you’ll go to the Define Resource panel (as was shown in Figure 76 on page 228) to enter the new resource name and description.

When you come back to the Resource Definition List panel, you can now select the new VECTOR resource to be part of the DB2LATE scheduling environment, as shown in Figure 80.

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAS                Resource Definition List          Row 1 to 5 of 5
Command ==>>> _____

Selection For Scheduling Environment DB2LATE

Action Codes: A=Add  S=Select  X=XREF  /=Menu Bar

Action  Resource Name      In Use  Resource Description
-----  -
_       CICS001            YES     CICS Subsystem
_       DB2A                YES     DB2 Subsystem
_       IMS1                YES     IMS Subsystem
_       PRIMETIME           YES     Peak Business Hours
s_     VECTOR              YES     Vector Processor

```

Figure 80. Resource Definition List panel

Having done that and exiting, you come back to the Modify a Scheduling Environment panel, as shown in Figure 81.

```

Scheduling-Environments  Notes  Options  Help
-----
IWMAPAJ                Modify a Scheduling Environment                Row 1 to 3 of 3
Command ==>>> _____

Scheduling Environment Name  DB2LATE_____ Required
Description . . . . . Offshift DB2 Processing_____

Action Codes: A=Add  D=Delete

Action  Resource Name      Required
-----  -
  _     DB2A              ON_____  DB2 Subsystem
  _     PRIMETIME       OFF_____ Peak Business Hours
  _     VECTOR          ON_____  Vector Processor

```

Figure 81. Modify a Scheduling Environment Panel

Now the VECTOR resource name is part of the DB2LATE scheduling environment. As always, you need to specify either ON or OFF under the **Required State**. In this case, we chose ON.

Summary of the two methods for creating new resources

You can create new resources in two ways:

- Directly from the main panel (“Creating a new resource directly from the main panel” on page 227). This method is useful when you want to create resources without actively working on one particular scheduling environment. Also, note that when you go to the Resource Definition List panel with this method, the Action Codes look like this:

```
Action Codes: A=Add  D=Delete  X=XREF  /=Menu Bar
```

There is no S=Select, because you are not actively working on any scheduling environment (and therefore there is no place for you to select resources to).

There is a D=Delete action code. This will be important when you are deleting resources, as in “Deleting a resource” on page 231.

- While creating or modifying scheduling environments (“Creating a new resource while working with a scheduling environment” on page 228). This method is useful when you want to create resources while in the midst of creating or modifying a scheduling environment. Note that when you go to the Resource Definition List panel with this method, the Action Codes (and the line above it) look like this:

```
Selection For Scheduling Environment DB2LATE
```

```
Action Codes: A=Add  S=Select  X=XREF  /=Menu Bar
```

The name of scheduling environment that you are currently working with appears on this panel, and there is a S=Select among the action codes. You can select the new resource to be a part of the scheduling environment immediately after creating it.

There is no D=Delete action code, because you *cannot* delete resources when you come the Resource Definition List panel this way. See “Deleting a resource” on page 231.

Showing all cross-references for a resource definition

You can check which specific scheduling environments use a given resource definition by entering an X in the action column next to the resource name, as

shown in Figure 82. (You can use either path to get to the Resource Definition List path, as discussed in “Summary of the two methods for creating new resources” on page 230.)

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAS          Resource Definition List          Row 1 to 4 of 4
Command ==>>> _____

Selection For Scheduling Environment DB2LATE

Action Codes: A=Add  S=Select  X=XREF  /=Menu Bar

Action  Resource Name  In Use  Resource Description
--      -
_       CICS001          YES     CICS Subsystem
_       IMS1             YES     IMS Subsystem
X_     PRIMETIME        YES     Peak Business Hours
_       VECTOR          YES     Vector Processor

```

Figure 82. Resource Definition List panel

This will take you to the Resource Cross-Reference of Scheduling Environments panel, as shown in Figure 83.

```

IWMAPAY          Resource Cross-Reference Of Scheduling Environments Row 1 to 2 of 2
Command ==>>> _____

Resource . . . . . : PRIMETIME
Description . . . . . : Peak Business Hours

Scheduling Environment Name  Description
IMSPRIME                    Prime IMS Processing
DB2LATE                     Offshift DB2 Processing

```

Figure 83. Resource Cross-Reference Of Scheduling Environments panel

Deleting a resource

Unlike adding a resource definition, there is only one path you can take to delete a resource definition: from the main Scheduling Environment Selection List panel (as discussed in “Summary of the two methods for creating new resources” on page 230).

Put the cursor under the Resources field at the top of the screen and hit ENTER. When the pop-up box appears, enter a 1, as was shown in Figure 74 on page 227. (In this pop-up, 1 is the only option.)

This will take you directly to the Resource Definition List panel. Enter a D on the Action line next to the resource definition you wish to delete, as shown in Figure 84 on page 232.

```

Resources  Notes  Options  XREF  Help
-----
IWMAPAR           Resource Definition List           Row 1 to 5 of 5
Command ==> _____

Action Codes: A=Add  D=Delete  X=XREF  /=Menu Bar

Action  Resource Name  In Use  Resource Description
--      -
--      CICS001         YES     CICS Subsystem
--      DB2A           YES     DB2 Subsystem
--      IMS1          YES     IMS Subsystem
--      PRIMETIME    YES     Peak Business Hours
D_      VECTOR          YES     Vector Processor

```

Figure 84. Resource Definition List panel

Note that you can only delete a resource definition that is not currently in use by any scheduling environment. If you attempted to delete the PRIMETIME resource in Figure 75 on page 228, you would receive an error message stating the resource cannot be deleted because a scheduling environment is using it. You must first check the cross-references for that resource definition (as in “Showing all cross-references for a resource definition” on page 230) and then go into every scheduling environment that uses it to delete those references.

Working with the guest platform management provider (GPMP) settings

On z/OS system images, the guest platform management provider (GPMP) is the interface between the z/OS workload manager (WLM) and the IBM zEnterprise Unified Resource Manager. The GPMP supports platform management functions and provides policy information to WLM about the platform-wide performance goals of workloads in which the z/OS system is participating.

Figure 85 shows the guest platform management provider (GPMP) settings panel:

```

GPMP-Settings  Notes  Options  Help
-----
                Guest Platform Management Provider (GPMP) Settings
Command ==> _____

Activate guest platform management provider: 1      1. NO
                                                2. YES

Names of systems to be excluded:

_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____

```

Figure 85. Guest Platform Management Provider Settings - sample panel

On this panel you can specify whether you want WLM to manage the guest platform management provider on each system of the sysplex. If activation is selected, WLM automatically starts each guest platform management provider and restarts it as needed. Otherwise, you must manually start and stop the guest platform management provider. By default, activation is not enabled.

In addition, you can specify the systems in the sysplex on which the guest platform management provider should not be started automatically.

Coordinating updates to a service definition

Since you can keep your service definition in either the WLM couple data set or in MVS data sets, it is possible to have multiple, uncoordinated updates. You should decide on a process for updating your service definition.

If you decide to work from the WLM couple data set, installing and extracting the service definition when you need to edit it, you should keep a copy in an MVS data set. The WLM application allows only one user to update an MVS data set at a time. If you do try to edit an MVS data set that is in use, you are locked out, and get a message saying the data set is in use.

If you select the Install option, and someone has done an install since you've extracted, the application displays a pop-up panel asking for confirmation. This way you can prevent inadvertent overwrites of the service definition. Figure 86 shows the Overwrite warning panel.

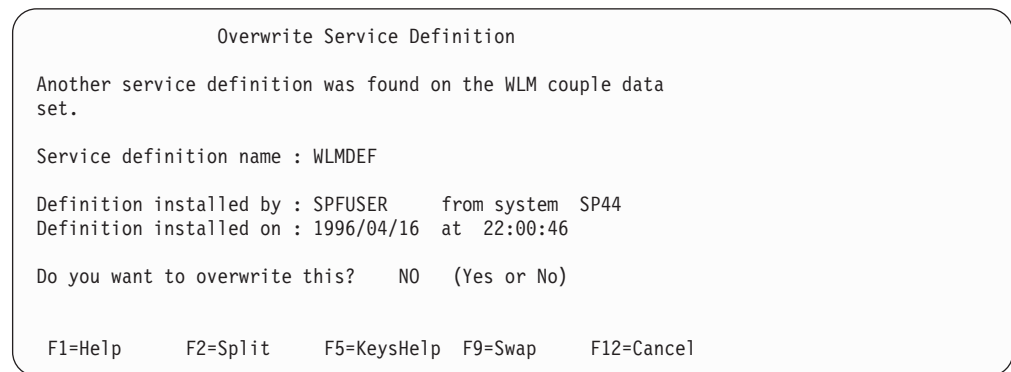


Figure 86. Overwrite Service Definition panel

Using the WLM couple data set

The service definition installed on the WLM couple data set is the *installed* service definition. The policy that you activate must be in the installed service definition.

Of course, before you can install your service definition on the WLM couple data set, you must first allocate one, and define it to the sysplex. You can allocate a WLM couple data set through a function in the WLM application, or you can use a utility. For information on how to allocate using the utility, see “Allocate a WLM couple data set” on page 160.

Allocating a WLM couple data set

For availability purposes, you should allocate both a primary and an alternate WLM couple data set.

You can allocate a new primary and alternate WLM couple data set from the application using the **Allocate couple data set using CDS values** function on the **Utilities** menu bar option of the Definition Menu. Figure 87 on page 234 shows the allocate panel that uses CDS values.

```

Allocate couple data set for WLM using CDS values
Command ==> _____
Sysplex name   SYSPLEX1 (Required)
Data set name  'GAILW.PCOUPLE.P1' (Required)
-----
Size parameters (optional):
Service policies . . 10 (1-99)
Workloads . . . . . 35_ (1-999)
Service classes . . 30_ (1-100)
Application
environments . . . . 50_ (1-3000)
Scheduling
environments . . . . 50_ (1-999)
SVDEF extensions . . 0__ (0-8092)
SVDCR extensions . . 0__ (0-8092)
SVAEA extensions . . 0__ (0-8092)
SVSEA extensions . . 0__ (0-8092)
Storage parameters:
Storage class . . . A
Management class . . _____
or
Volume . . . . . _____
Catalog data set?  _ (Y or N)

```

Figure 87. Allocate couple data set using CDS values panel

This function primes the size parameters with values from the current couple data set. This way, you can ensure that the new couple data set is at least as large as the current one. If the new one is smaller than the current one, you will not be able to use SETXCF to make this data set available to the sysplex. (You will receive a series of messages starting with IXC255I UNABLE TO USE DATA SET.) You may need to increase one or more of these primed values if you have added new objects to your service definition since the last time you installed it.

If there is no current WLM couple data set, use the **Allocate couple data set** function on the **Utilities** menu. This function fills in the size parameters based on the service definition you are about to install. It gets the values from the ISPF PDS containing the service definition. Figure 88 shows the allocate panel that uses service definition values.

```

Allocate couple data set for WLM
Sysplex name   SYSPLEX2 (Required)
Data set name  'KIRSTEN.PCOUPLE.P1' (Required)
-----
Size parameters (optional):
Service policies . . 10 (1-99)
Workloads . . . . . 35_ (1-999)
Service classes . . 30_ (1-100)
Application
environments . . . . 50_ (1-3000)
Scheduling
environments . . . . 50_ (1-999)
Storage parameters:
Storage class . . . A
Management class . . _____
or
Volume . . . . . _____
Catalog data set?  _ (Y or N)

```

Figure 88. Allocate couple data set panel using service definition values

If you have defined your service definition, the application fills in the number of policies, workloads, service classes, application environments, and scheduling

environments in that definition. These actual values are the best estimate of what this service definition requires when installed on the couple data set. If you define too large a number, it can use up a lot of space.

If you are replacing couple data sets already in use, the new data sets must be at least as large as the existing ones. You need to run a special sizing utility to determine their size. Use this method to ensure that the new couple data set is at least as large as the current one, and, in most cases, can accommodate any new objects you might have added to the service definition:

1. First use the **Allocate couple data set** option as a quick way of displaying the values required for the service definition you have just created or modified. Write down these values and exit the panel without allocating a couple data set.
2. Then use the **Allocate couple dataset using CDS values** function to obtain a panel primed with the current couple data set values.
3. Take the maximum of the values from steps 1 and 2 and enter these values on the panel. Complete the couple data set allocation from this panel.

Once you have allocated the WLM couple data sets, you need to make them available for use in the sysplex. If you are making couple data sets available for the first time, see “Make a WLM couple data set available to the sysplex for the first time” on page 163. If you are replacing couple data sets already in use, see “Make a newly formatted couple data set available to the sysplex” on page 164.

Installing and extracting a service definition

You can install and extract the service definition using the **Install** and **Extract** functions on the **Utilities** menu bar option of the Definition Menu.

Installing the service definition overwrites any service definition previously installed on the WLM couple data set.

When you extract the service definition, a copy remains on the WLM couple data set until the next install.

Using MVS data sets

If you have multiple people updating the service definition, or if you want to work with more than one service definition, you can work in multiple MVS data sets. For example, if you want to work on the service definition for next year's SLA, you can keep it in an MVS data set.

If you specify **Read saved definition**, you can choose from a list of data sets already created. The list is made up of all data sets your user IDs has worked with (the list is built based on use).

Restricting access to your service definition

To restrict access to the service definition stored in both the MVS data set and the WLM couple data set, you can use the same resource control facility as you do for an MVS data set. For information about how to restrict access to the WLM application functions, see “Restricting access to the WLM service definition” on page 156.

Activating a service policy

You can activate a service policy from the ISPF application, as well as from the operator console. To activate a policy from the operator console, you issue the **VARY WLM** command. For more information about issuing the **VARY** command, see “Migration activities” on page 156. To activate a policy from the application, choose the **Utilities** option from the menu bar on the Definition menu.

When you choose the activate option, the application extracts the service definition information from the WLM couple data set, and displays the Policy Selection List panel, as shown in Figure 89. From the list, you select the service policy that you want to activate.

If you have been editing your service definition and want your changes to take effect, you have to install the changed service definition, and then activate the service policy.

```
Policy Selection List          ROW 1 TO 3 OF 3

The following is the current Service Definition installed on the WLM
couple data set.

Name . . . . : SLA1993

Installed by : USERID    from system SYSTEM
Installed on : 1996/04/01 at 11:30

Select the policy to be activated with "/"

Sel Name      Description                                ---Last Change---
  _ BACKUPS   During weekly backups                     User      Date
  _ DAYTIME   Policy from 7:00am to 5:00 pm                 NORTH    1996/05/11
  _ NIGHT     Late night batch window                             NORTH    1996/05/11
***** END OF DATA *****

Command ==> _____
```

Figure 89. Policy Selection List panel to activate a service policy

Printing in the application

The print function on the selection lists in the WLM application prints the information into an ISPF list data set. You can process the ISPF list data set just as you would any other ISPF list data set. For more information about using list data sets, see *Interactive System Productivity Facility (ISPF) User's Guide*.

The **Print as GML** function under the **File** action bar choice of the Definition menu prints the complete service definition to a data set in GML format. The data set can be formatted using **SCRIPT/VS**. For more information, see *DCF SCRIPT/VS User's Guide*.

Note: To use **SCRIPT/VS**, you must have Document Composition Facility (DCF) Version 3, or higher, installed.

Browsing definitions

The **Browse** function is available on selection lists in the application, and is similar to the print function. **Browse** prints information into a temporary data set, and displays it.

Figure 90 shows the browse on the BAT_T service class from the service class selection list.

```
Browse                               Line 00000000 Col 001 072
***** TOP OF DATA *****
* Service Class BAT_T - Batch class T

Created by user BERKEL on 1996/10/07 at 11:29:05
Base last updated by user BERKEL on 1996/10/07 at 11:29:05

Base goal:

# Duration Imp Goal description
- - - - -
1          5 80% complete within 00:30:00.000

***** END OF DATA *****

Command ==>
```

Figure 90. Browse function from the Service Class Selection List

Using XREF function to view service definition relationships

To view the relationship between any two definitions (service class, workload resource group etc), you can use the XREF function. From any Modify panel in the application, you can use the XREF function to determine an definition's relationship to another.

For example, on the Modify Service Class panel, you may want to know whether a service policy overrides the defined goal. You use the XREF function to check whether its goals are overridden in a service policy. From the Modify Service Class panel, you can also check which subsystem types reference that service class in the classification rules. If there are some that do reference it, the application displays a pop-up selection list where you can browse the Rules referencing that service class. Figure 91 shows an example of the pop-up displayed for an Xref by Subsystem for a service class.

```
Service Class Reference ROW 1 TO 1 OF 1

These subsystem types refer to the service class.

Action Codes: 4=Browse

Action Name Description
_ STC IBM-defined subsystem type
***** BOTTOM OF DATA *****
```

Figure 91. Service Class Subsystem Xref panel

WLM application messages

IWMAM040 Unexpected error, RC=xx, RSN=xxxxxxxx.

Explanation: The application has detected an unexpected error.

In the message text:

xx The return code.

xxxxxxxx

 The hexadecimal reason code.

System action: The requested operation is not performed.

Programmer response: Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide the text of this message.

Module: Workload manager (WLM)

IWMAM041 WLM couple data set is unavailable.

Explanation: Workload manager (WLM) was unable to find the WLM couple data set. It is possible that it does not exist. Or it may exist, but it has not been defined to the sysplex.

System action: The requested operation is not performed.

Programmer response: Either:

- Make the couple data set available to the sysplex by updating the DATA keyword of the COUPLExx parmlib member or issuing the SETXCF command.
- Allocate the WLM couple data set.

Module: Workload manager (WLM)

IWMAM042 Extract failed, no service definition was found on the WLM couple data set.

Explanation: Workload manager (WLM) was unable to find a service definition on the WLM couple data set.

System action: The extract is not performed.

Programmer response: Ensure that the service definition has been properly installed on the WLM couple data set.

Module: Workload manager (WLM)

IWMAM043 Unable to obtain storage, RSN=xxxxxxxx.

Explanation: Workload manager (WLM) was unable to obtain enough storage to complete the operation.

In the message text:

xxxxxxxx

 The hexadecimal reason code.

System action: The requested operation is not performed.

Programmer response: Increase the region size and repeat the operation. If this does not help, search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide the text of this message.

Module: Workload manager (WLM)

IWMAM044 Install failed, service definition is not valid. Validation reason code: xxxx, Validation offset: yyyy

Explanation: The service definition you are trying to install is not valid.

xxxx is the validation reason code. yyyy is the validation offset (a hex offset into the iwmszzz data structures where the problem exists).

System action: The install is not performed.

Programmer response: For further explanation of this error, see the topic "Application Validation Reason Codes" in *z/OS MVS Programming: Workload Management Services*. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide the text of this message.

Module: Workload manager (WLM)

IWMAM046 Errors were found during validation of the service definition. The install has failed.

Explanation: The errors listed on the previous panel (shown when you attempted to install) have prevented the installation of the service definition.

System action: The install is not performed.

Programmer response: Correct the errors and retry. To capture a list of the errors, go back to the error panel by either attempting to install the service definition again or by validating the service definition (use the "Validate definition" utility as described in "Using the menu bar on the Definition Menu" on page 203). From that panel, you can capture a list of the errors by selecting "Save listing" on the "File" menu bar option.

Module: Workload manager (WLM)

IWMAM047 WLM couple data set is too small to hold the service definition.

Explanation: The service definition being installed is too large to fit into the WLM couple data set.

System action: The requested action is not performed.

Programmer response: Re-allocate the WLM couple data set with a larger size, as described in "Allocate a WLM couple data set" on page 160 and "Make a WLM couple data set available to the sysplex for the first time" on page 163.

Module: Workload manager (WLM)

IWMAM050 Exceeded the maximum number of attempts to read the service definition.

Explanation: A failure in reading the service definition from the WLM couple data set has occurred repeatedly.

System action: The requested operation is not performed.

Programmer response: Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide the text of this message.

Module: Workload manager (WLM)

IWMAM051 Access was denied to the WLM couple data set.

Explanation: The user does not have appropriate RACF authority to the WLM couple data set.

System action: The requested operation is not performed.

Programmer response: Verify that the user should be authorized and have the RACF administrator give the user appropriate access. For information about how to restrict access to the couple data set, see "Restricting access to the WLM service definition" on page 156.

Module: Workload manager (WLM)

IWMAM052 The service definition functionality level (LEVELxxx) is not compatible with the WLM ISPF application level (LEVELyyy). To extract a service definition or to activate a policy, the WLM ISPF application and the MVS system must be at the same level as the service definition.

Explanation: The service definition in the WLM couple data set uses functions that are not compatible with this level of the WLM ISPF application

System action: The requested operation is not performed.

Programmer response: Use the WLM ISPF application on a system that is compatible with the functionality level of the service definition.

IWMAM054 • IWMAM077

Module: Workload manager (WLM)

IWMAM054 Failure in ISPF: ISPF error information.

Explanation: An error occurred using the application.

System action: The requested operation is not performed.

Programmer response: The ISPF error information may provide a clue as to how to overcome the problem. If not, search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide the text of this message.

Module: Workload manager (WLM)

IWMAM055 Extract failed, service definition is not valid. Validation reason code: *xxxx*, Validation offset: *yyyy*

Explanation: The service definition on the WLM couple data set is not valid. The data set may be corrupted.

xxxx is the validation reason code. *yyyy* is the validation offset.

System action: The extract is not performed.

Programmer response: If the data set is corrupted, try restoring it from backups.

Module: Workload manager (WLM)

IWMAM058 Install failed. WLM couple data set has not been reallocated for use with this OS/390 Release. Reallocate the WLM couple data set. Refer to the Migration Chapter in "MVS Planning: Workload Management."

Explanation: The WLM couple data set upon which you are attempting to install a service definition is not formatted for your service definition.

System action: The system does not install the service definition.

Programmer response: Re-allocate the WLM couple data set for the service definition.

Module: Workload manager (WLM)

IWMAM072 The service definition was not read due to a mismatch between the service definition PDS (LEVELxxx) and the WLM ISPF application (LEVELyyy). To read the service definition PDS, restart the WLM ISPF application at level LEVELxxx or higher.

Explanation: The service definition in the PDS uses functions that are not compatible with this level of the WLM ISPF application.

System action: The requested operation is not performed.

Programmer response: Use the WLM ISPF application on a system that is compatible with the functionality level of the service definition.

Module: Workload manager (WLM)

IWMAM077 Unable to use *datasetname* for service definition data, member *membername* has an unrecognized format.

Explanation: WLM cannot use the *datasetname* definition data as a WLM service definition. WLM has determined that member *membername* in the PDS contains information that is not recognized by WLM. The incorrect member contains key and name information that does not match what WLM expects. Note that the member shown is the first member that was found to be invalid. It is possible that others are invalid, or that the entire data set is corrupted.

System action: The requested operation is not performed.

Programmer response: If you are using a downlevel version of the WLM ISPF, it may no longer be possible to open an ISPF service definition data set written by a higher level WLM Administrative Application. (See "Migrating to a new z/OS release with an existing service definition" on page 155.) Otherwise, the data set may be corrupted. If you cannot see an obvious problem in the data set, contact the IBM Support Center.

Module: Workload manager (WLM)

IWMAM098 The service definition was not read due to a mismatch between the service definition XML (LEVELxxx) and the WLM ISPF application (LEVELyyy). To read the service definition XML restart the WLM ISPF application at LEVELxxx or higher.

Explanation: The service definition in the data set uses functions that are not compatible with this level of the WLM ISPF application.

System action: The requested operation is not performed.

Programmer response: Use the WLM ISPF application on a system that is compatible with the functionality level of the service definition.

Module: Workload manager (WLM)

IWMAM099 The service definition *datasetname* cannot be used because it has an incompatible format. Probable cause: the service definition was last modified by a WLM ISPF application at level *level* which is higher than the level of this application.

Explanation: The service definition in the data set uses an ISPF table layout that is not compatible with this level of the WLM ISPF application.

System action: The requested operation is not performed.

Programmer response: Either use the WLM ISPF application on a system that is compatible with the ISPF table layout of the service definition, or use XML export and import to load the service definition on the backlevel system.

Module: Workload manager (WLM)

IWMAM313 No more than 2047 report classes may be defined.

Explanation: It is not possible to define more than 2047 report classes.

System action: The requested operation is not performed.

Programmer response: Do not use more than 2047 report classes.

Module: Workload manager (WLM)

Chapter 22. Using the z/OS Management Facility (z/OSMF) to administer WLM

This information briefly describes the **Workload Management task** in the z/OS Management Facility (z/OSMF).

Overview of the z/OSMF workload management task

The workload management task in z/OS Management Facility (z/OSMF) provides a browser-based user interface that you can use to manage z/OS workload manager (WLM) service definitions and provide guidelines for WLM to use when allocating resources. Specifically, you can define, modify, view, copy, import, export, and print WLM service definitions. You can also install a service definition into the WLM couple data set for the sysplex, activate a service policy, and view the status of WLM on each system in the sysplex.

Key functions of the Workload Management task in z/OSMF

The following information describes some of the key functions available in the Workload Management task in z/OSMF:

Display list of service definitions.

The Workload Management task provides a list of the WLM service definitions that have been defined in z/OSMF along with history information (for example, when the service definition was installed or modified), messages, and user activity. The list of service definitions is retrieved from the service definition repository, which refers to the directory in the z/OSMF data file system in which the data for the Workload Management task is stored.

Work with multiple service definitions.

In the Workload Management task, you can work with multiple service definitions simultaneously. To do so, open the service definitions with which you want to work in its own View, Modify, Copy, or Print Preview tab. You can also define multiple service definitions at the same time by opening several New tabs.

Install service definitions.

The Workload Management task provides features that you can use to install a service definition into the WLM couple data set for the z/OSMF host sysplex.

Extract the installed service definition.

The Workload Management task automatically extracts the service definition that is installed in the WLM couple data set for the z/OSMF host sysplex and stores it in the service definition repository so that you can view it, modify it, or activate one of its service policies.

Import and export service definitions.

The Workload Management task provides features that you can use to import a service definition from or export a service definition to your local workstation or a sequential data set on the z/OSMF host system. The exported service definition is formatted so that it can be opened with the z/OS WLM Administrative Application (also called the WLM ISPF application).

Provide table view and print preview of the service definition.

The Workload Management task provides two views of a service definition:

- **Table View.** The table view displays the parts of the service definition as tables. You can display the table view by opening the service definition in the **New, View, Modify, or Copy** tab. If you open the service definition in the **New, Modify, or Copy** tab, you can modify the service definition. In the *View* tab, you cannot modify the service definition.
- **Print Preview.** The print preview presents the service definition in HTML format and allows you to select which parts of the service definition you want to preview or print. You can display the print preview by opening the service definition in the **Print Preview** tab.

Activate service policies.

In the Workload Management task, you can specify which policy to activate when you install a service definition or you can activate a service policy that is defined in the service definition currently installed in the WLM couple data set for the sysplex.

Preview service policies with overrides applied.

The Workload Management task allows you to preview an HTML formatted version of the service policy with overrides applied. The HTML formatted service policy contains the information that would be included in the policy if it were activated. To preview a service policy, open the policy in the **Print Preview** tab.

View the sysplex status.

The Workload Management task provides an HTML formatted view (**WLM Status** tab) of the same data that is retrieved when you issue the `D WLM,SYSTEMS` command on the z/OS console. Specifically, the **WLM Status** tab displays the status of WLM on each system in the sysplex, and lists details about the installed service definition and the active service policy.

Define settings.

The Workload Management task provides a shared location (**Settings** tab) where you can specify for how long to keep the service definition history and define the code page, time zone, and backup sequential data set for the sysplex. You can also enable consistency checking between z/OSMF and the WLM couple data set, and indicate whether you want the Workload Management task to display or suppress information messages.

Actions that require the Workload Management task to interact with the sysplex are limited to the sysplex in which the z/OSMF host system is a member. Such actions include installing a service definition, activating a service policy, viewing the sysplex status, and so on. If you want to interact with another sysplex, z/OSMF must be installed on a system in that sysplex and you must log into that z/OSMF instance. You can use the service definition import and export functions to copy a service definition from one z/OSMF instance to another z/OSMF instance.

To display the Workload Management task, expand the Performance category in the z/OSMF navigation area and select **Workload Management**.

Figure 92 on page 245 shows the main panel for the Workload Management task. The **Overview** tab serves as the launch point for the actions that your user ID is authorized to access within the Workload Management task. To start using the

Workload Management task, select one of the actions listed in the **Overview** tab.

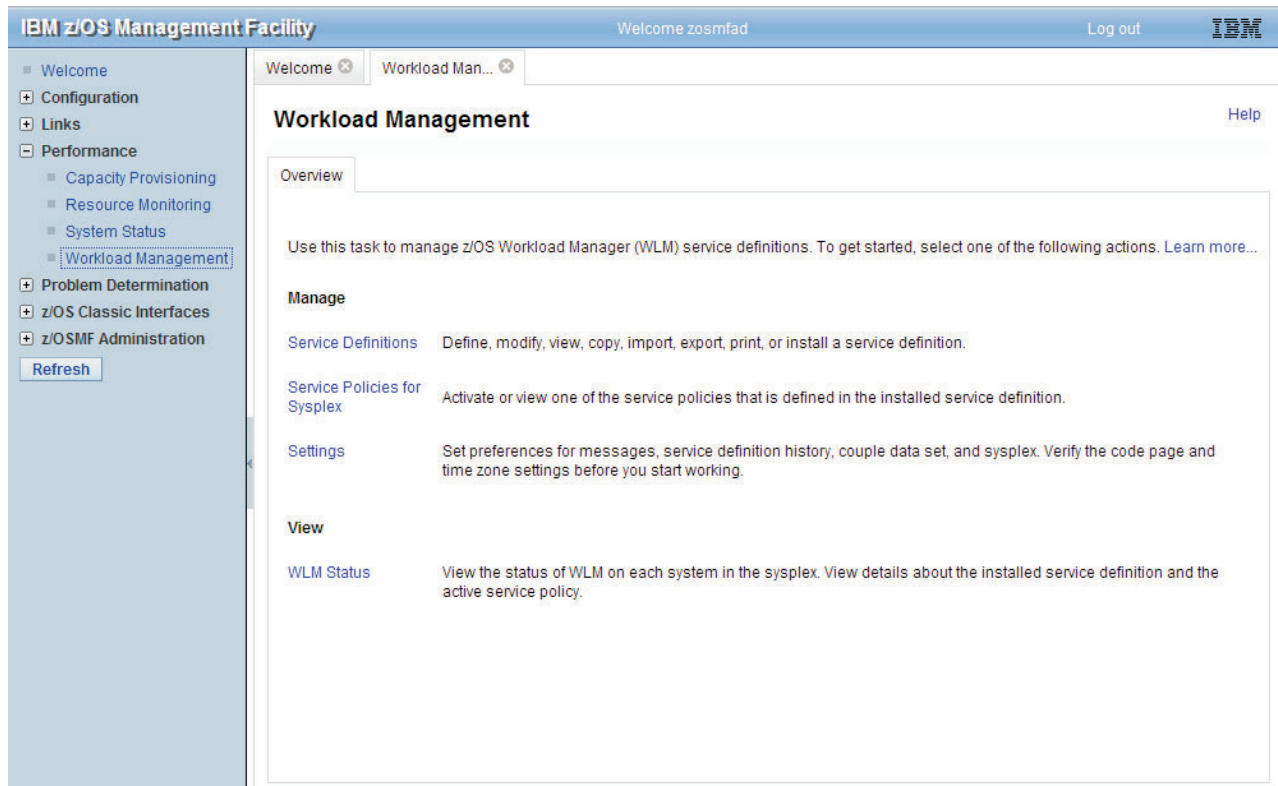


Figure 92. z/OS Management Facility - Overview Panel

For more information about the configuration of the Workload Management task in z/OSMF, see *IBM z/OS Management Facility Configuration Guide*.

Appendix A. Customizing the WLM ISPF application

This information explains how to customize your WLM application to:

- Customize the WLM application libraries.
If you have renamed or changed your IPCS/WLM library names, then you can use the IWMAREX1 exit to specify your library names.
- Customize the WLM application data sets.
If you would like to allocate the application data sets with your storage management policies, use IWMAREX2.
- Add the WLM application as an option on your ISPF menu.
If you plan to use the WLM application frequently, you can add the WLM application as an option on your ISPF application.
- Move pop-up windows.
- Customize the keylists.

Specifying the exits

To start the WLM application, you use a TSO/E REXX exec IWMARIN0. IWMARIN0 concatenates the IPCS/WLM data sets, allocates some data sets required for a service definition, and invokes the application panels. If EXITS is specified on the EXEC statement, IWMARIN0 uses the exits in the specified data set.

Table 18 shows the return codes from IWMARIN0.

Table 18. Return codes from IWMARIN0

Return Code	Explanation
4	Not in ISPF. Application cannot be started.
8	Unexpected keyword (parameter) on WLM application invocation. Unexpected keyword(parameter) is: <i>keyword</i> . <i>keyword</i> represents the keyword (parameter) from the command invocation.
12	Unexpected error occurred when calling installation exit IWMAREX1. One of the following: <ul style="list-style-type: none">• Installation exit IWMAREX1 must exist in <i>data-set name</i> for the WLM application to run. <i>data-set name</i> represents the data set that must contain the installation exit. Check to make sure the data set contains IWMAREX1.• Installation exit IWMAREX1 must exist in the current concatenation order for the WLM application to run.
16	Unexpected keyword (parameter) from WLM exit IWMAREX1. Please check coding in WLM installation exit IWMAREX1 for incorrect keyword (parameter): <i>parameter</i> . <i>parameter</i> represents the keyword (parameter) that is incorrect.

Table 18. Return codes from IWMARIN0 (continued)

Return Code	Explanation
20	Unexpected RC="rc" from TSO ALTLIB ISPF LIBDEF for <i>data-set-name</i> The WLM application cannot be started due to ALTLIB or LIBDEF failures for <i>data-set-name</i> . See specific REXX messages for the names of the data sets which failed.
24	ALTLIB failed during attempt to find installation exit data set. WLM application can not be started. TSO ALTLIB RC=xx. xx represents return code from TSO ALTLIB service.

Coding the WLM exits

The exit stubs are shipped in SYS1.SBLSCLI0. If you want to customize the application with the exits, create a data set for your exits, copy IWMAREX1 or IWMAREX2 into the data set, and modify the exits with your options. When you create the data set for your exits, specify the same data set characteristics as SYS1.SBLSCLI0.

You specify the exit data set on the TSO/E EXEC statement when you start the WLM application. Specify the fully qualified data set name containing the exit. For example, suppose you created data set IPCS.EXIT5, and coded IWMAREX1. To start the WLM application with the exit, you specify:

```
EX 'SYS1.SBLSCLI0(IWMARIN0)' 'EXIT(IPCS.EXIT5)'
```

IWMARIN1

If you have previously allocated the required data sets, either in a logon procedure, or in a CLIST, you can use IWMARIN1 to start the WLM application. To use IWMARIN1, specify:

```
EX 'SYS1.SBLSCLI0(IWMARIN1)'
```

Customizing the WLM application libraries — IWMAREX1

WLM provides the IWMAREX1 exit to specify the IPCS/WLM libraries names. If you have renamed or customized the following IPCS/WLM libraries, use IWMAREX1 to specify your names.

Table 19. WLM Libraries

Library	Content
SYS1.SBLSCLI0	Application REXX code data set
SYS1.SBLSKEL0	Application skeleton data set
SYS1.SBLSPNL0	Application panel data set
SYS1.SBLSTBL0	Application keylists and commands data set
SYS1.SBLSMSG0	Application messages data set

If you have renamed them in your installation, use IWMAREX1 to set up the allocations.

IWMAREX1 is a REXX routine for specifying installation-customized data sets required for starting the WLM ISPF application.

If you have renamed or changed the WLM/IPCS data sets in your installation, use IWMAREX1 to set up the allocations.

Processing

IWMAREX1 is called from the IWMARIN0 REXX exec.

Parameters

IWMAREX1 has the following parameters:

REXXDS

The application REXX code data set.

SKELDS

The application skeleton data set.

PANELDS

The application panel data set.

TABLEDS

The application tables (keylist and commands) data set.

MESSAGEDS

The application messages data set.

Example

Suppose the IPCS/WLM application resides in SYS1.IPCS.SBLSCLI0, and you have renamed your IPCS/WLM libraries to:

```
SYS1.IPCS.SBLSMSG0  
SYS1.IPCS.SBLSPNL0  
SYS1.IPCS.SBLSKEL0  
SYS1.IPCS.SBLSTBL0
```

Suppose you have created your exit in a data set called WLM.EXITS. You code IWMAREX1 in the following way:

```
/* REXX */  
queue 'REXXDS(SYS1.IPCS.SBLSCLI0)'  
queue 'SKELDS(SYS1.IPCS.SBLSKEL0)'  
queue 'PANELDS(SYS1.IPCS.SBLSPNL0)'  
queue 'TABLEDS(SYS1.IPCS.SBLSTBL0)'  
queue 'MESSAGEDS(SYS1.IPCS.SBLSMSG0)'  
Exit 0
```

To start the WLM application with the exit, you specify:

```
EX 'SYS1.IPCS.SBLSCLI0(IWMARIN0)' 'EXIT(WLM.EXITS)'
```

Customizing the WLM application data sets — IWMAREX2

WLM provides the IWMAREX2 exit to specify:

- The application recovery data set and data set characteristics.
- The service definition data set and data set characteristics.

You should use this exit to allocate data sets according to your installation's storage management policies. You can not change any of the following TSO/E ALLOCATE parameters in IWMAREX2:

- Data set name (DA)
- Record format (RECFM)
- Logical record length (LRECL)
- Data set organization (DSORG)

Unless you code otherwise in this exit, the WLM application uses the following data set information as the defaults:

```
UNIT(SYSDA)
TRACKS SPACE(15,15)
```

Processing

This exit is called from the IWMARZAL REXX exec.

Parameters

IWMAREX2 has the following parameters:

ARDSDIR

Specifies the number of 256 byte records to be allocated for the application recovery data set.

ARDSOPTS

Specifies the options for the application recovery data sets, with the following sub-parameters:

UNIT(*xxxx*)

Specifies the unit type for the TSO ALLOCATE command.

STORCLAS

The SMS storage class.

MGMTCLAS

The SMS management class.

DATACLAS

The SMS data class.

To determine what to specify for unit, check which UNIT type is coded for the TSO ALLOCATE command in your installation.

ARDSSPACE

Specifies the options to allocate a data set. The options are:

SPACE(*quantity,(increment)*)

TRACKS

CYLINDERS

BLOCKS(*value*)

PDDSDIR

Specifies the number of 256 byte records to be allocated for the print service definition data set.

PDDSOPTS

Specifies the options for the print service definition data set, with the following sub-parameters:

UNIT(*xxxx*)

Specifies the unit type for the TSO ALLOCATE command.

STORCLAS

The SMS storage class.

MGMTCLAS

The SMS management class.

DATACLAS

The SMS data class.

To determine what to specify for unit, check which UNIT type is coded for the TSO ALLOCATE command in your installation.

PDDSSPACE

Specifies the options to allocate a print definition data set. The options are:

SPACE(*quantity,(increment)*)**TRACKS****CYLINDERS****BLOCKS(*value*)****SDDSDIR**

Specifies the number of 256 byte records to be allocated for the service definition data set.

SDDSOPTS

Specifies the options for the service definition data sets, with the following sub-parameters:

UNIT(*xxxx*)

Specifies the unit type for the TSO ALLOCATE command.

STORCLAS

The SMS storage class.

MGMTCLAS

The SMS management class.

DATACLAS

The SMS data class.

SDDSSPACE

Specifies the options to allocate a data set. The options are:

SPACE(*quantity,(increment)*)**TRACKS****CYLINDERS****BLOCKS(*value*)****XMLDSOPTS**

Specifies the options for the service definition XML data sets, with the following sub-parameters:

UNIT(*xxxx*)

Specifies the unit type for the TSO ALLOCATE command.

STORCLAS

The SMS storage class.

MGMTCLAS

The SMS management class.

DATACLAS

The SMS data class.

XMLDSSPACE

Specifies the options to allocate a service definition XML data set. The options are:

SPACE(*quantity*,(*increment*))

TRACKS

CYLINDERS

BLOCKS(*value*)

Examples

The following examples show some uses of the IWMAREX2 exit:

- Suppose you want to specify that the UNIT type for your TSO allocate commands in your installation is type SYSDS. In exit IWMAREX2, you specify the following:

```
/* REXX */
'ARDSOPTS(UNIT(SYSDS))'
Exit 0
```

- Suppose you want to specify the service definition data sets as SMS managed data sets in the standard storage class, and in the NOMIG management class. In exit IWMAREX2, you specify the following:

```
/* REXX */
queue 'ARDSOPTS(STORCLAS(STANDARD) MGMTCLAS(NOMIG))'
queue 'ARDSSPACE(SPACE(10,10) TRACKS)'
Exit 0
```

Adding WLM as an ISPF menu option

To add the WLM application as an option on your ISPF primary panel, you should make a copy of the ISPF primary option menu - ISR@PRIM. You need to add some information to the processing section of the panel. What you add depends on whether you use IWMARIN0 or IWMARIN1 to start the application.

If you use IWMARIN0, specify:

```
WLM,'CMD(EXEC "SYS1.SBLSCLI0(IWMARIN0)") NEWAPPL(IWMP) PASSLIB'
```

If you use the WLM exits together with IWMARIN0, specify:

```
WLM,'CMD(EXEC "SYS1.SBLSCLI0(IWMARIN0)" "EXIT(ISPF.EXIT)" ) NEWAPPL(IWMP) PASSLIB'
```

If you use IWMARIN1, specify:

```
WLM,'CMD(EXEC "SYS1.SBLSCLI0(IWMARIN1)") NEWAPPL(IWMP) PASSLIB'
```

Make sure you concatenate the library containing your customized primary panel before any others in your logon procedure or CLIST.

Example: Adding WLM as an option on your ISPF menu

Figure 93 shows the two lines added to a copy of the ISPF primary panel. In the example, IWMARIN1 is specified. The two lines are highlighted in the figure.

```
----- ISPF/PDF PRIMARY OPTION MENU -----
%OPTION ==>_ZCMD
%
% 0 +ISPF PARS - Specify terminal and user parameters +USERID -
% 1 +BROWSE - Display source data or output listings +TIME -
% 2 +EDIT - Create or change source data +TERMINAL -
% 3 +UTILITIES - Perform utility functions +PF KEYS -
% 4 +FOREGROUND - Invoke language processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO/E command or CLIST
% 7 +DIALOG TEST - Perform dialog testing
% 8 +LM UTILITIES- Perform library management utility functions
% C +CHANGES - Display summary of changes for this release
% W +WLM - WLM administrative application
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
 .HELP = ISR00003
 &ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
 &ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
 &ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
 VPUT (ZHTOP,ZHINDEX) PROFILE
)PROC
 &ZSEL = TRANS( TRUNC (&ZCMD, '.')
 0, 'PANEL(ISPOPTA)'
 1, 'PGM(ISRBRO) PARM(ISRBRO01)'
 2, 'PGM(ISREDIT) PARM(P,ISREDM01)'
 3, 'PANEL(ISRUTIL)'
 4, 'PANEL(ISRFPA)'
 5, 'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
 6, 'PGM(ISRPTC)'
 7, 'PGM(ISRYXDR) NOCHECK'
 8, 'PANEL(ISRLPRIM)'
 C, 'PGM(ISPTUTOR) PARM(ISR00005)'
 W, 'CMD(EXEC "SYS1.SBLSCLI0(IWMARIN1)") NEWAPPL(IWMP) PASSLIB'
 T, 'PGM(ISPTUTOR) PARM(ISR00000)'
 ' ', ' '
 X, 'EXIT'
 *, '?' )
 &ZTRAIL = .TRAIL
)END
```

Figure 93. Example of adding WLM as an option on the ISPF menu

Moving pop-up windows

If you would like to customize the placement of the pop-up windows in the WLM application, you can use a manual ISPF function. You must place the cursor anywhere on the active window frame and press Enter. ISPF acknowledges the window move request by displaying WINDOW MOVE pending message. Then place the cursor where you want the upper left corner of the pop-up placed. press Enter a second time, and the pop-up is moved to the new location.

Note: The placement lasts only for the duration of the session. If you exit the application, your changes are lost.

There are some other options for moving pop-up windows. For more information about them, see *ISPF Dialog Management Guide and Reference*.

Customizing the keylists

The WLM application uses a set of keylists which you can customize to your purposes. To edit a keylist, type:

```
KEYLIST
```

on the command line from any panel in the WLM application. The application displays the keylist utility panel, as shown in Figure 94. From this panel, select the keylist you would like to work with, and choose an action from **Options** on the menu bar.

```

Options  Change Keylists
-----
ISPKLUP      Keylist Utility for IWMP      ROW 1 TO 9 OF 9
Command ==> _____

Enter keylist name _____ OR

Select one keylist name from the list below:
Select  Keylist  T  -
-      KEYSBRP  S
-      KEYSWRK  S
-      KEYS001  S
-      KEYS002  S
-      KEYS01H  S
-      KEYS01P  S
-      KEYS01S  S *** CURRENTLY ACTIVE KEYLIST ***
-      KEYS02A  S
-      KEYS02B  S
-      KEYS02P  S
***** END OF DATA *****

```

Figure 94. Keylist Utility panel

The keylists and the type of panels on which they are used are:

Table 20. Keylist names and usage descriptions

Keylist	Function
KEYS001	Non-scrollable, non pop-up
KEYS002	Scrollable, non pop-up
KEYS01P	Non-scrollable, pop-up
KEYS02P	Scrollable, pop-up
KEYS01H	All help panels
KEYS01S	Non-scrollable, non pop-up, no PF4=RETURN
KEYS02A	Scrollable, non pop-up, PF1=HELPD, PF10=ACTIONS.
KEYS02B	Scrollable, non pop-up, PF1=HELPD, PF10=LEFT, PF11=RIGHT.
KEYSBRP	Browse panel
KEYSWRK	"Working..." panel

Appendix B. CPU capacity table

The tables presented at www.ibm.com/servers/resourcelink/lib03060.nsf/pages/srminde?OpenDocument show the unweighted CPU service units per second by CPU model. You use this information to define your minimum and maximum capacity for a resource group.

Also, for the latest information about the processor version codes and SRM constants, see the online documentation available at the same location.

If you plan to use these constants for purposes other than those suggested in this information, observe the following limitations:

- Actual customer workloads and performance may vary. For a more exact comparison of processors, see the internal throughput rate (ITR) numbers in *Large Systems Performance Reference (LSPR)*.
- CPU time can vary for different runs of the same job step. One or more of the following factors might cause variations in the CPU time: CPU architecture (such as storage buffering), cycle stealing with integrated channels, and the amount of the queue searching (see *z/OS MVS System Management Facilities (SMF)*).
- The constants do not account for multiprocessor effects within logical partitions. For example, a logical 1-way partition in an S/390 9672, Model RX3, has 1090 service units per second, while a 10-way partition on the same machine has 839.3 service units per second.

Using SMF task time

For installations with no prior service data, the task time reported in SMF type 4, 5, 30, 34, and 35 records can be converted to service units using the tables in Appendix B, "CPU capacity table."

Examples of resource groups

- To give a department a "dedicated 4381" amount of capacity, you can specify the following resource group:

Name 4381-91E

Description
Capacity of a 4381

Capacity Minimum
309

Capacity Maximum
309

- To assure half the capacity of a 9672 Model Y96 system, with 9 physical CPs, to a service class, specify the following:

Name HALFSYS3

Description
Preserve half of SYS3 (9672 Mod Y96)

Capacity Maximum
24129

Since the 9672 Model Y96 has 5362 raw CPU service units per second, with 9 physical CPs, $0.5 \times 9 \times 5362$ is 24129. The 24129 is captured service units allocated to the address spaces in the resource group.

Appendix C. Return codes for the IWMINSTL sample job

This information describes the return codes issued by the IWMINSTL sample job. The Install Definition Utility is shipped as member IWMINSTL in SYS1.SAMPLIB. You can use the IWMINSTL job to install a WLM service definition or to activate a WLM policy without having to use the ISPF WLM application.

Table 21 describes the return codes that are issued by IWMINSTL. An accompanying message text is written to the job output listing.

Table 21. Return codes from IWMINSTL

Return code	Message text and explanation
0	Successful execution.
104	The service definition was not processed due to a mismatch between the WLM address space level (current.wlm.level) and the level of this utility (IWMARIDU.level). Use the correct level of the IWMARIDU utility.
204	GetServiceDefinition Unable to use (your.definition.dataset) for service definition data. A table has an unrecognized format.
208	GetServiceDefinition Unable to use (your.definition.dataset), data set is in use.
212	GetServiceDefinition The service definition was not read due to a mismatch between the service definition PDS and the WLM Install Definition Utility. The service definition PDS has a higher functionality level than the current version of the WLM Install Definition Utility. Once a service definition was updated by a WLM instance with a higher functionality level it can no longer be updated by a WLM instance with lower functionality level.
216	GetServiceDefinition The service definition was not opened due to an ISPF dialog error.
304	InstallServiceDefinition Install failed, service definition has no name. No service definition name was specified. To fix the error, do the following: <ul style="list-style-type: none">• Specify a valid service definition name for IWMINSTL parameter SVDEFPDS.• Verify in the IWMINSTL JCL, whether the SVDEF DD statement exists.
308	InstallServiceDefinition Install failed, no workloads are defined.
312	InstallServiceDefinition Install failed, no service classes are defined.
316	InstallServiceDefinition Install failed, no service policies are defined.

Table 21. Return codes from IWMINSTL (continued)

Return code	Message text and explanation
320	<p>InstallServiceDefinition Install failed, access was denied to the WLM couple data set. The user ID does not have update authority to the RACF resource MVSADMIN.WLM.POLICY in the FACILITY class.</p>
324	<p>InstallServiceDefinition InstallServiceDefinition Install failed, service definition was modified</p> <p>Every WLM service definition contains a service definition ID. An attempt was made to install a service definition with a different ID than the service definition ID that is installed. A WLM service definition ID consists of the following:</p> <ul style="list-style-type: none"> • The name of the service definition • A timestamp when the service definition was installed • A user ID that installed the service definition • The system name from which the installation was done <p>Use the FORCE=Y parameter for the WLM install definition utility to install a service definition with a different ID. The default is FORCE=N.</p>
336	<p>InstallServiceDefinition Install failed, failure in WLM bridge layer. Additional messages regarding this error might be written to the job output listing.</p>
404	<p>ActivateServicePolicy The service policy was not activated due to an ISPF dialog error.</p>
408	<p>ActivateServicePolicy Activate failed, activation in progress on another system.</p>
412	<p>ActivateServicePolicy Activate failed, one or more systems were unable to activate the policy.</p>
416	<p>ActivateServicePolicy Activate failed, (POLNAME) was not found on the WLM couple data set.</p>
420	<p>ActivateServicePolicy Activate failed, failure in WLM bridge layer. Additional messages regarding this error might be written to the job output listing.</p>
516	<p>SaveServiceDefinition The service definition was not opened due to an ISPF dialog error.</p>

Appendix D. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome>).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS V2R2 ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out

punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element

The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element

The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the

default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

*** indicates an optional syntax element that is repeatable**

The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain 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.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (<http://www.ibm.com/software/support/systemsz/lifecycle/>)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, and service names may be trademarks or service marks of others.

Workload management terms

active service policy

The service policy that determines workload management processing.

application environment

A group of application functions requested by a client that execute in server address spaces.

application owning region (AOR)

In a CICSplex configuration, a CICS region devoted to running applications.

automatic control

One of two distinct methods of managing application environments. Under automatic control, the name of the startup JCL procedure has been defined for an application environment, giving workload management the ability to automatically start server address spaces. Contrast with *manual control*.

channel path identifier

The channel subsystem communicates with I/O devices by means of a channel path between the channel subsystem and devices. A CHPID is a value assigned to each channel path of the System z that uniquely identifies that path. Up to 256 CHPIDs are supported for each channel subsystem.

CICSplex

A configuration of interconnected CICS systems in which each system is dedicated to one of the main elements of the overall workload. See also *application owning region*, *file owning region*, and *terminal owning region*.

classification rules

The rules workload management and subsystems use to assign a service class and, optionally, a report class to a work request. A classification rule consists of one or more of work qualifiers such as subsystem type, subsystem instance, userid, accounting information, transaction name, transaction class, source LU, netid, and LU name.

compatibility mode

Prior to z/OS V1R3 a mode of processing, in which the IEAIPSxx and IEAICSxx

parmlib members determine system resource management. See also *goal mode*.

control region

The main storage region that contains the subsystem work manager or subsystem resource manager control program.

couple data set

A data set created through the XCF couple data set format utility. The data set is shared by MVS systems in a sysplex. There are several types of couple data sets for different purposes. See also *WLM couple data set*.

CPU service units

A measure of the task control block (TCB) execution time multiplied by an SRM constant which is CPU model dependent. See also *unweighted CPU service units per second*, and *service unit*.

delay monitoring services

The workload management services that monitor the delays encountered by a work request.

distributed data facility (DDF)

An optional feature that allows a DB2 application to access data at other DB2s and at remote relational database systems that support IBM's Distributed Relational Database Architecture™ (DRDA).

duration

The length of a service class performance period in service units.

dynamic alias management

A service definition option — when enabled, workload management will dynamically reassign parallel access volume aliases to help work meet its goals and to minimize IOS queueing.

enclave

A transaction that can span multiple dispatchable units (SRBs and tasks) in one or more address spaces and is reported on and managed as a unit.

execution velocity

A service goal naming the rate at which you expect work to be processed for a given service class or a measure of the

acceptable processor and storage delays while work is running.

fold qualifier names

When defining classification rules, the "Fold qualifier names" option, when set to the default Y, means that the qualifier names will be folded to upper case as soon as you type them in and then press Enter. If you set this option to N, then the qualifier names will remain in the case they are typed in.

goal mode

A mode of processing where the active service policy determines system resource management.

guest platform management provider (GPMP)

An optional suite of applications that is installed in specific z/OS, Linux, and AIX® operating system images to support platform management functions. For example, the guest platform management provider collects and aggregates performance data for virtual servers and workloads. Users view these reports through the ensemble-management Hardware Management Console (HMC).

Hardware Management Console (HMC)

A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the CPC's Support Element (SE). On a zEnterprise 196, using the Unified Resource Manager on the HMCs/SEs, personnel can also create and manage an ensemble.

HMC Accepted acronym for Hardware Management Console.

hypervisor

A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, zVM, and PowerVM®.

IBM System z

Family name. After first use can be System z.

IBM System z Application Assist Processor (zAAP)

A specialized processor that provides a Java™ execution environment, which enables Java-based Web applications to be integrated with core z/OS business applications and backend database systems.

IBM z Enterprise Unified Resource Manager (Unified Resource Manager)

Utilize on "first use". Then can use z Enterprise Unified Resource Manager or short form Unified Resource Manager. See "Unified Resource Manager".

IBM System z Integrated Information Processor (zIIP)

A specialized processor that provides computing capacity for selected data and transaction processing workloads, and for selected network encryption workloads.

importance level

the degree of importance of a service goal relative to other service class goals, in five levels: lowest, low, medium, high, highest.

intranode management network (INMN)

A private 1000BASE-T Ethernet network operating at 1 Gbps that is required for the Unified Resource Manager to manage the resources within a single zEnterprise node. The INMN connects the Support Element (SE) to the z196 and to any attached zBX.

I/O priority management

A service definition option — when enabled, I/O priorities will be managed separately from dispatching priorities, according to the goals of the work.

I/O service units

A measure of individual data set I/O activity and JES spool reads and writes for all data sets associated with an address space.

installation

A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

installed service definition

The service definition residing in the WLM couple data set for WLM.

logical unit (LU)

In *VTAM*, the source and recipient of data transmissions. Data is transmitted from one logical unit (LU) to another LU. For example, a terminal can be an LU, or a CICS or IMS system can be an LU.

LU Logical unit.

LU name

The second level of the source LU name after the "." for fully qualified names.

LU 6.2 session

A session that is initiated by *VTAM** programs on behalf of a logical unit (LU) 6.2 application program, or a session initiated by a remote LU in which the application program specifies that the *VTAM* programs are to control the session by using the *APPCCMD* macro instruction.

manual control

One of two distinct methods of managing application environments in goal mode. Under manual control, the name of the startup JCL procedure has not been defined for an application environment. The installation must therefore manually start server address spaces when needed. Contrast with *automatic control*.

masking

Using a % for a single character replacement in classification rules. See also *wild carding*.

MVS image

The system-id of any *MVS* system included in the sysplex as it appears in *SYS1.PARMLIB* member.

performance administration

The process of defining and adjusting workload management goals and resource groups based on installation business objectives.

performance block

A piece of storage containing workload management's record of execution delay information about work requests.

performance management

The process workload management uses to decide how to match resources to work according to performance goals and processing capacity.

performance period

A service goal and importance level assigned to a service class for a specific duration. You define performance periods for work that has changing performance requirements as work consumes resources.

policy See *service policy*.

relational database management system (RDBMS)

A relational database manager that supports *SAA*.

report class

A group of work for which reporting information is collected separately. For example, you can have a report class for information combining two different service classes, or a report class for information on a single transaction.

resource

When used as part of a scheduling environment, a resource is an abstract element that can represent an actual physical entity (such as a peripheral device), or an intangible quality (such as a certain time of day). A resource is listed in a scheduling environment along with a required state of ON or OFF. If the corresponding resource state on a given system matches the required state, then the requirement is satisfied for that resource.

resource group

An amount of processing capacity across one or more *MVS* images, assigned to one or more service classes.

scheduling environment

A list of resource names along with their required states. If an *MVS* image satisfies all of the requirements in the scheduling environment associated with a given unit of work, then that unit of work can be assigned to that *MVS* image. If any of the requirements are *not* satisfied, then that unit of work *cannot* be assigned to that *MVS* image.

server address space

Any address space that does work on behalf of a transaction manager or a resource manager. For example, a server address space could be a CICS AOR, or an IMS control region.

service administration application

The online ISPF application used by the service administrator to specify the workload management service definition.

service class

A group of work which has the same performance goals, resource requirements, or business importance. For workload management, you assign a service goal and optionally a resource group to a service class.

service coefficient

A value that specifies which type of resource consumption should be emphasized in the calculation of service rate. The types of resource consumption are CPU, IOC, MSO, and SRB.

service definition

A definition of the workloads and classification rules in an installation. The definition includes workloads, service classes, systems, resource groups, service policies, and classification rules.

service level administrator

The user role introduced by workload management whose main task is to make sure overall installation operation is consistent with performance goals and objectives.

service level agreement (SLA)

A written agreement of the information systems (I/S) service to be provided to the users of a computing installation.

service policy

A named set of performance goals workload management uses as a guideline to match resources to work. See also *active service policy*.

service request block (SRB) service units

A measure of the SRB execution time for both local and global SRBs, multiplied by an SRM constant which is CPU model dependent.

service unit

The amount of service consumed by a work request as calculated by service definition coefficients and CPU, SRB, I/O, and storage service units.

single-system sysplex

A sysplex in which only one MVS system is initialized as part of the sysplex. In a

single-system sysplex, XCF provides XCF services on the system, but does not provide signalling services between MVS systems. See also *multi-system sysplex*, *XCF-local mode*, and *monoplex*.

source LU

A fully qualified two level name separated by a ".", where the first level is the network id and the second is the LU name, OR merely a single LU name. See also *LU name*.

storage service units

A measure of the central storage page frames multiplied by 1/50 of the CPU service units. The 1/50 is a scaling factor designed to bring the storage service component in line with the CPU component.

subsystem instance

1) For application environments, a unique combination of subsystem type (as specified in the service definition for an application environment) and subsystem name (as specified by the work manager subsystem when it connects to workload management). 2) For classification, a work qualifier used to distinguish multiple instances of a subsystem.

subsystem work manager

An address space defined in the SYS1.PARMLIB member as SUBSYS=nnn.

terminal owning region (TOR)

A CICS region devoted to managing the terminal network.

Unified Resource Manager

Full name: IBM zEnterprise Unified Resource Manager. Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

unweighted CPU service units per second

The unweighted service units per second of task or SRB execution time. This measure is CPU-model dependent, but is independent of the values of the service coefficients.

velocity

A service goal naming the rate at which you expect work to be processed for a given service class or a measure of the acceptable processor and storage delays while work is running.

wild carding

The use of an asterisk (*) as a multiple character replacement in classification rules. See also *masking*.

WLM couple data set

A type of couple data set that is created through the XCF couple data set format utility for the WLM function. The data set contains the service definition information.

workload

A group of work to be tracked, managed and reported as a unit. Also, a group of service classes.

workload management mode

The mode in which workload management manages system resources on an MVS image. Prior to z/OS V1R3 this mode could be either *compatibility mode*, or *goal mode*. Starting with z/OS V1R3, compatibility mode has been removed.

work qualifier

An attribute of incoming work. Work qualifiers include: subsystem type, subsystem instance, userid, accounting information, transaction name, transaction class, source LU, netid, and LU name.

work request

A piece of work, such as a request for service, a batch job, an APPC, CICS, or IMS transaction, a TSO LOGON, or a TSO command.

zAAP Approved acronym for IBM System z Application Assist Processor (zAAP). See full name.

zIIP Approved acronym for IBM System z Integrated Information (zIIP). See full name.

Index

A

- access
 - restricting 156
- accessibility 259
 - contact IBM 259
 - features 259
- accounting information
 - nesting 69
 - qualifier 69
- action field 198
- active service policy
 - definition 267
- adjusting velocity goals 56
- administer WLM
 - with z/OS Management Facility 243
- administration application
 - definition 270
- alias management, dynamic 105
- APPC/MVS scheduler (ASCH) 69
- application
 - action field 198
 - classification groups 218
 - command line 198
 - commands 198
 - create a group panel 218
 - create workload panel 206
 - customizing keylists 254
 - definition menu 200
 - function keys 199
 - scrollable area 197
 - starting 249, 252, 253
 - using the menu bar 196
 - workload selection list 206
 - workloads 206
- application environment
 - definition 267
- application environments 131
 - authorizing 135
 - CB (WebSphere Application Server) 125
 - changing the definition of 134
 - DB2 125
 - defining 125
 - getting started with 125
 - handling error conditions in 134
 - IWEB 125
 - making changes to servers 134
 - managing 132
 - overview 11
 - selecting server limits for 128
 - SOM 125
 - specifying to workload management 126
 - using operator commands for 133
- application owning region
 - definition 267
- ASCH (APPC/MVS scheduler)
 - overview of work 64
 - work qualifiers supported by 69
- assistive technologies 259

- automatic control
 - definition 267
- average response time
 - limit 50
 - limits 50

B

- balancing of WLM managed batch initiators 20, 168
- batch initiator balancing 20, 168
- batch initiator management 168
- batch initiators 21
- business importance
 - definition 7

C

- capacity definition 28
- capacity, minimum and maximum 41
- CB
 - overview of work 64
- CB (WebSphere Application Server)
 - application environments 125
 - work qualifiers supported by 69
- changing goal types in performance periods 58
- channel subsystem priority queuing 176
- CHPID
 - definition 267
- CICS (customer information control system)
 - overview of work 64
 - work qualifiers supported by 69
- CICSplex
 - definition 267
- classification
 - defining rules 61
 - defining the order 85
 - inheritance 85
 - nesting 63, 69
 - qualifiers supported 67
 - supporting subsystems 63
- classification rules
 - creating for a subsystem type 213
 - definition 267
- client accounting information
 - qualifier 70
- client IP address
 - qualifier 70
- client transaction name
 - qualifier 71
- client user ID
 - qualifier 71
- client workstation name
 - qualifier 71
- collection name
 - qualifier 71
- compatibility mode
 - definition 267

- connection type
 - qualifier 71
- contact
 - z/OS 259
- control region
 - definition 267
- correlation information
 - qualifier 72
- couple data set
 - allocating 160
 - calculating the size of 160
 - definition 267
 - increasing the size of 162
 - installing service definition 233
 - restricting access to 156
 - SEXTXCF command 163
 - updating COUPLExx 163
- COUPLExx member
 - DATA keyword 163
- coupling facility
 - defining a structure 169, 173
- CPSM environment 20
- CPU protection 50, 110
- CPU service units
 - definition 267, 270
- customer information control system (CICS) 69

D

- DB2
 - application environments 125
 - overview of work 64
 - work qualifiers supported by 69
- DB2 distributed data facility
 - environment 21
- DDF (distributed data facility)
 - overview of work 64
 - work qualifiers supported by 69
- DDF environment 21
- defining application environments 125
- defining capacity 28
- defining scheduling environments 137
- defining velocity goals 56
- discretionary goal 50
- discretionary goal management
 - migration considerations 166
- discretionary goals
 - using 57
- dispatch mode 23
- distributed data facility (DDF) 69
- duration
 - definition 51
 - in a performance period 51
- dynamic alias management 105
 - definition 267
 - migration considerations 166
- dynamic channel path management 175
 - coupling facility structure 173

E

- enclave
 - definition 267
- Enterprise Storage Server 105
- execution velocity
 - calculation 104
 - definition 267
 - with I/O priority management 104

F

- fold qualifier names
 - definition 268
 - option explained 213
- functionality level 158

G

- goal
 - definition 49
- goal mode
 - definition 268
- goal types 18
- goals, performance
 - See also* performance goals
 - defining 53
- GPMG (guest platform management provider) 147
- group capacity 29
- guest platform management provider (GPMP) 147

H

- Hardware Management Console (HMC)
 - definition 268
- heterogeneous report class 99
- HiperDispatch mode 23
- HMC
 - definition 268
- homogeneous report class 99
- honor priority 111
- Honor Priority 50
- hypervisor
 - definition 268

I

- I/O priority
 - enabling 104
 - management 104
- I/O Priority Group 50
- I/O priority management
 - definition 268
- I/O priority queueing
 - defining 33
 - dynamic 33
- I/O service units
 - definition 268
- I/O storage management 25
- importance
 - definition 268
- importance levels
 - in performance periods 59

- IMS (information management system)
 - overview of work 64
 - work qualifiers supported by 69
- information management system (IMS) 69
- inheritance
 - in classification rules 85
- initiators, batch 21
- Install Definition Utility
 - member IWMINSTL 154
 - member IWMSSDEF 154
- installation exit
 - IWMAREX1 248
 - IWMAREX2 250
- installed service definition
 - definition 268
- intelligent resource director
 - channel subsystem priority queuing 176
 - coupling facility structure 173
 - dynamic channel path management 175
 - LPAR CPU Management 175
 - making it work 178
- intranode management network (INMN)
 - definition 268
- IO protection 111
- IRD
 - coupling facility structure 173
- IWEB
 - application environments 125
 - overview of work 64
 - work qualifiers supported by 69
- IWMAM040 message 238
- IWMAM041 message 238
- IWMAM042 message 238
- IWMAM043 message 238
- IWMAM044 message 238
- IWMAM046 message 239
- IWMAM047 message 239
- IWMAM050 message 239
- IWMAM051 message 239
- IWMAM052 message 239
- IWMAM054 message 240
- IWMAM055 message 240
- IWMAM058 message 240
- IWMAM072 message 240
- IWMAM077 message 240
- IWMAM098 message 241
- IWMAM099 message 241
- IWMAM313 message 241
- IWMAREX1 installation exit 248
- IWMAREX2 installation exit 250
- IWMINSTL 154
- IWMINSTL, return codes 257
- IWMSSDEF 154

J

- JES (job entry subsystem)
 - overview of work 64
 - work qualifiers supported by 69
- JES2 (job entry subsystem 2)
 - overview of work 64
 - work qualifiers supported by 69
- JES2 batch initiators 21

- JES3 (job entry subsystem 3)
 - overview of work 64
 - work qualifiers supported by 69
- JES3 batch initiators 21
- job entry subsystem (JES) 69

K

- keyboard
 - navigation 259
 - PF keys 259
 - shortcut keys 259

L

- LDAP
 - work qualifiers supported by 69
- LPAR clustering
 - coupling facility structure 173
- LPAR CPU management 175
- LPAR weight management 175
- LSFM
 - overview of work 64
 - work qualifiers supported by 69
- LU 6.2 session
 - definition 269
- LU name
 - definition 269
 - qualifier 72

M

- managing batch initiators 168
- managing resource states 139
- manual control
 - definition 269
- masking
 - definition 269
- masking notation
 - examples 90
- maximum capacity 41
- memory
 - limit 41
- memory limit 41
- menu bar
 - definition menu 203
 - on selection list 197
- migration
 - creating service definition for the first time 153
 - discretionary goal management considerations 166
 - dynamic alias management considerations 166
 - from pre-version 5 153
 - multisystem enclaves considerations 166
 - overview 153
 - velocity considerations 165
 - with an existing service definition 155
- minimum capacity 41
- mixed-release sysplex 158
- mode
 - definition 271

- modifications of
 - transaction server management 113
- MQ (MQSeries Workflow)
 - overview of work 64
 - work qualifiers supported by 69
- MQSeries Workflow (MQ) 69
- MSO coefficient
 - using 103
- multiple periods 18
- multisystem enclaves
 - coupling facility structure 169
 - migration considerations 166

N

- navigation
 - keyboard 259
- nesting
 - accounting information 86
 - subsystem parameter 86
- netid
 - qualifier 72
- NETV
 - work qualifiers supported by 69
- non z/OS partition CPU
 - management 27
- notation
 - start position 91
- notepad
 - updating 33
 - using 33
- Notices 263

O

- OMVS (z/OS UNIX System Services)
 - overview of work 64
 - work qualifiers supported by 69

P

- package name
 - qualifier 72
- parallel access volume 105
- partitioned data set
 - restricting access to 156
- PERFORM
 - qualifier 72
- performance administration
 - definition 4
- performance goal
 - definition 49
- performance goals
 - defining 53
 - definition 7
 - discretionary 7
 - execution velocity 7
 - response-time 7
- performance group
 - qualifier 72
- performance management
 - definition 4
 - with guest platform management
 - provider (GPMP) 147
- performance period
 - definition 269

- performance period (*continued*)
 - maximum number 51
- performance periods
 - using 58
 - using importance levels in 59
- periods, multiple 18
- plan name
 - qualifier 72
- policy overrides
 - defining 36, 210
 - examples 36
 - resource group association 36
 - resource group capacity 36
- printing
 - service definition 236
 - service policy 236
- priority
 - qualifier 73
- procedure name
 - qualifier 73
- process name
 - qualifier 73
- processor model
 - service units 255
 - task/SRB execution time 255
- protection
 - CPU 110
 - IO 111
 - storage 109
- protection options for critical work
 - defining 109

Q

- qualifier
 - accounting information 69
 - client accounting information 70
 - client IP address 70
 - client transaction name 71
 - client user ID 71
 - client workstation name 71
 - collection name 71
 - connection type 71
 - correlation information 72
 - grouping 93
 - LU name 72
 - nesting 86
 - netid 72
 - package name 72
 - PERFORM 72
 - performance group 72
 - plan name 72
 - priority 73
 - procedure name 73
 - process name 73
 - scheduling environment name 73
 - subsystem collection name 74
 - subsystem instance 74
 - subsystem parameter 74
 - sysplex name 75
 - system name 76
 - transaction class 76
 - transaction name 77
 - user ID 78
 - zEnterprise service class from a
 - Unified Resource Manager
 - performance policy 79

- qualifier names
 - folding 213, 268
- qualifiers
 - definition 61

R

- RACF
 - restricting access to WLM service
 - definition 156
- report class
 - assigning 99
 - definition 99, 269
 - heterogeneous 99
 - inhomogeneous 99
- resource
 - definition 269
- resource group
 - defining 41
 - definition 269
 - limitations 41
 - maximum capacity 41
 - minimum capacity 41
 - overview 8
 - removing a service class from 211
- resource requirements 137
 - definition 7
- resource states, managing 139
- response time goals, system
 - determining 53
- response time with percentile
 - limit 50
- restricting access to WLM service
 - definition 156
- return codes
 - sample job IWMINSTL 257

S

- SAF
 - restricting access to WLM service
 - definition 156
- samples included in velocity goals 56
- scheduling environment
 - definition 269
- scheduling environment name
 - qualifier 73
- scheduling environments
 - associating with incoming work 143
 - defining 137
 - getting started with 137
 - managing resource states 139
 - overview 12
 - specifying to workload
 - management 138
 - using special characters 138
- security server
 - restricting access to WLM service
 - definition 156
- selection lists
 - in classification rules 218
 - status line 197
- sending comments to IBM xiii
- server management, transaction 113
- service class
 - definition 49, 270

- service class (*continued*)
 - maximum number 49
 - SYSOTHER 89
 - SYSSTC 89
 - SYSSTC1-SYSSTC5 89
 - SYSTEM 88
 - system-provided 88
 - service coefficient
 - defaults 103
 - defining 102
 - definition of term 270
 - service definition
 - base 14
 - contents 33
 - creating for the first time 153
 - defining 33
 - definition 5, 270
 - hierarchy 14
 - installing 154, 233
 - printing 236
 - printing as GML 236
 - restricting access to 156
 - storing in MVS PDS 195
 - storing in WLM couple data set 195
 - service level agreement (SLA)
 - definition 270
 - service policy 36
 - activating 236
 - create service policy panel 205
 - defining 6, 35
 - defining overrides 210
 - definition 270
 - in service definition 35
 - overview 14
 - printing 236
 - service unit
 - definition 270
 - processor model 255
 - task/SRB execution time 255
 - SETXCF command
 - example 163
 - shark 105
 - shortcut keys 259
 - SOM
 - application environments 125
 - overview of work 64
 - work qualifiers supported by 69
 - source LU
 - definition 270
 - SRB (service request block)
 - execution time
 - processor model 255
 - service units 255
 - SRB service units
 - definition 270
 - start position
 - using 91
 - started task control (STC) 69
 - started tasks
 - defining goals for 57
 - servers 57
 - started tasks (STC)
 - defining classification rules 95
 - defining service classes 95
 - STC (started task control)
 - overview of work 64
 - work qualifiers supported by 69
 - STC (started tasks)
 - defining classification rules 95
 - defining service classes 95
 - storage coefficient
 - using 103
 - storage management
 - I/O 25
 - storage protection 109
 - storage service units
 - definition 270
 - substring notation
 - examples 91, 92, 93
 - subsystem collection name
 - qualifier 74
 - subsystem instance
 - definition 270
 - qualifier 74
 - subsystem parameter
 - qualifier 74
 - subsystem type
 - creating 218
 - deleting 218
 - modify rules for 213
 - subsystem types
 - in classification 67
 - summary of changes
 - z/OS V2R2 xvi
 - SYS_ 138
 - SYS1.PARMLIB
 - member IEAOPT 24
 - SYS1.SAMPLIB
 - member IWMINSTL 154
 - member IWMSSDEF 154
 - SYSH
 - overview of work 64
 - work qualifiers supported by 69
 - SYSOTHER
 - service class 89
 - sysplex couple data set
 - upgrading 154
 - sysplex name
 - qualifier 75
 - SYSSTC
 - service class 89
 - SYSSTC1-SYSSTC5
 - service class 89
 - SYSTEM
 - service class 88
 - system name
 - qualifier 76
 - system response time goals
 - determining 53
 - System z Application Assist
 - Processor 185
 - System z Integrated Information
 - Processor (zIIP) 191
- T**
- task
 - processor model 255
 - TCP
 - work qualifiers supported by 69
 - terminal owning region (TOR)
 - definition 270
 - transaction class
 - qualifier 76
 - transaction name
 - qualifier 77
 - transaction server management,
 - modifications of 113
 - TSO
 - overview of work 64
 - work qualifiers supported by 69
- U**
- Unified Resource Manager
 - definition 270
 - Unified Resource Manager performance
 - policy, service class
 - qualifier 79
 - user ID
 - qualifier 78
 - user interface
 - ISPF 259
 - TSO/E 259
 - using
 - group capacity 29
 - z/OS Management Facility 243
- V**
- velocity
 - definition 271
 - formula 52
 - limit 50
 - migration considerations 165
 - velocity goals
 - adjusting 56
 - defining 56
 - samples included in 56
- W**
- WebSphere Application Server (CB) 69
 - wild carding
 - definition 271
 - wildcard notation 90
 - examples 90
 - WLM couple data set
 - allocating 160
 - calculating the size of 160
 - definition 271
 - increasing the size of 162
 - installing service definition 233
 - restricting access to 156
 - SEXTXCF command 163
 - updating COUPLExx 163
 - work environments 17
 - work qualifier
 - definition 271
 - work request
 - definition 271
 - workload
 - definition 271
 - in service definition 39
 - workload balancing 19
 - definition 5
 - Workload Management task
 - overview 243

Z

- z/OS Management Facility 243
- z/OS UNIX System Services (OMVS) 69
- zAAP
 - definition 268
- zAAP (System z Application Assist Processor) 185
- zIIP
 - definition 268
- zIIP (System z Integrated Information Processor) 191



Product Number: 5650-ZOS

Printed in USA

SC34-2662-05

