CPExpert™

WLM Component Volume 1

Computer Management Sciences, Inc
6076-D Franconia Road
Alexandria, Virginia 22310-1756
voice: (703) 922-7027
fax: (703) 922-7305
www.cpexpert.com

CPExpert is a trademark of Computer Management Sciences, Inc.

This manual applies to the WLM Component of **CPExpert**[tm], a proprietary software product of Computer Management Sciences, Inc., Alexandria, Virginia, United States of America.

The information in this document is subject to change. Comments or suggestions are welcome, and to the extent practicable, will be incorporated in revisions to this document. Please send all comments, suggested new rules, suggested changes to existing rules, suggestions for improvement to the software or documentation, or any other advice to:

Computer Management Sciences, Inc.
6076-D Franconia Road
Alexandria, Virginia 22310-1756
(703) 922-7027 FAX: (703) 922-7305
www.cpexpert.com

DISCLAIMER

The advice, recommendations, or otherwise contained in this document represent information generally available in the public domain, as contained in vendor manuals, published in articles or papers, presented at professional conferences, or otherwise commonly accepted in the professional community. Neither Computer Management Sciences, Inc. nor its representatives make representations or warranties with respect to the applicability or application of any advice, recommendations, or otherwise, contained in this document or in any results from applying the CPExpert software, to any particular computer system or computer installation.

TRADEMARKS

CPExpert is a trademark of Computer Management Sciences, Inc. IBM, MVS/370, MVS/SP, MVS/XA, MVS/ESA, Enterprise System/3090, Netview, PR/SM, Processor Resource/System Manager, Hiperspace, and ES/9000 are trademarks of the IBM Corporation. SAS, SAS/OR, and SAS/STAT are trademarks of the SAS Institute Inc. MXG is a trademark of Merrill Consultants. MICS is a trademark of Legent Corporation.

COPYRIGHT INFORMATION

©Copyright 1994, Computer Management Sciences, Inc.
All rights reserved. Printed in the United States of America.

This licensed work is confidential and propriety, and is the property of Computer Management Sciences, Inc. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the specific written authorization of Computer Management Sciences, Inc.

Preface

In 1994, IBM began delivering the *Workload Manager* as a part of the new MVS/ESA SP5.1 operating system release. The Workload Manager provides a radical departure from earlier IBM operating system software.

- With earlier IBM system software, performance objectives were achieved by providing MVS with specific parameters which provided either direct control or general guidance to MVS in how system resources were to be used by workloads. The controls were *resource oriented*, in that users provided MVS with detailed values which MVS used as its control mechanism. Installation personnel were responsible for associating detailed controls with specific workloads (for example, users could provide CPU dispatching priorities, memory constraints, multiprogramming level constraints. etc.).

- With the Workload Manager, users are no longer in direct control of system resources and how those resources are allocated to workloads. Rather, MVS attempts to allocate or re-allocate system resources to meet **performance goals** as specified by users. Installation personnel provide MVS with end-user goals (e.g., average transaction response)

The Workload Manager is a new concept and new software. Consequently, there will be an industry-wide "learning curve" as installations gain experience using the Workload Manager to meet performance goals.

The WLM Component of CPExpert can help reduce the learning curve by (1) automatically analyzing how well the Workload Manager meets performance goals, (2) analyzing performance data to detect when performance goals are not achieved, (3) identifying the reasons performance goals were not achieved, and (4) suggesting alternatives to improve performance.

Additionally, the WLM Component can help reduce the learning curve by providing comprehensive documentation about Workload Manager concepts, algorithms, and operational features.

For those performance analysts who have a good grasp of Workload Manager concepts, constraints, and analysis techniques, the WLM Component can **significantly** help reduce the time required to perform daily analysis of system performance. This is because the WLM Component reports and analyzes periods when *service classes fail to achieve their performance goals*. Since only problem situations are reported by the WLM Component, performance analysts can focus on these problems and not waste time analyzing other areas.

How to use this manual

This document describes how to use the WLM Component of CPExpert to analyze major constraints to improved performance of your computer system. The manual is organized into two volumes, consisting of four sections and one appendix.

Section 1 (contained in Volume 1) provides an introduction to the WLM Component. This section is organized into four chapters. Most of this section can be reviewed for general information.

> Chapter 1 provides a brief overview of IBM's Workload Manager.
>
> Chapter 2 provides an overview of the WLM Component of CPExpert.
>
> Chapter 3 describes the sources of data used by the WLM Component to analyze performance of your system.
>
> Chapter 4 describes the performance data bases which CPExpert can use to analyze performance.

Section 2 (contained in Volume 1) describes how to provide guidance variables to the WLM Component using the prefix.CPEXPERT.SOURCE(WLMGUIDE) PDS member. The instructions in this chapter will be important when you install the WLM Component and at any time the guidance variables need to be changed. This section is organized into two chapters.

> Chapter 1 describes how to specify data selection and results presentation guidance variables. The instructions in this chapter will be important if you wish to select specific measurement periods for analysis, select or exclude service classes, or to alter the default way in which the WLM Component presents results.
>
> Chapter 2 describes how to specify threshold control variables to guide the WLM Component in its analysis of system performance. The instructions in this chapter will be important if you wish to alter the defaults provided with the WLM Component.

Section 3 (contained in Volume 1) describes how to use the WLM Component to analyze performance. The instructions in this section will be followed each time you execute the WLM Component. This section is organized into two chapters.

> Chapter 1 provides detailed instructions on executing the WLMCPE Module to analyze the performance of your system. The instructions in this chapter will be important each time that the WLM Component is executed.

Chapter 2 contains a checklist for executing the WLM Component.

Section 4 (contained in Volume 1) discusses considerations for analyzing the Workload Manager. This section will be important to understand how the Workload Manager operates and how the results from CPExpert should be evaluated. This section is organized into five chapters.

Chapter 1 presents an overview of the Workload Manager concepts.

Chapter 2 describes how the performance of subsystem transactions (i.e., CICS transactions or IMS transactions) is managed by the Workload Manager, and discusses the performance information available from subsystems.

Chapter 3 discusses some performance implications of the internal logic of the Workload Manager.

Chapter 4 describes the performance data which are used by CPExpert to analyze system performance from the perspective of the Workload Manager.

Chapter 5 highlights some of the factors that must be considered when analyzing performance based upon workload data collected and recorded by RMF.

Appendix A contains a detailed description of each rule that results in a finding based upon the WLM Component analyzing performance of your system. You may wish to briefly review the rules in this appendix to appreciate the problems that are encountered in different installations. However, it is not necessary to read all of the rules. It is necessary only to read the rules that are identified by the reports produced from the WLMCPE Module based on an analysis of performance at **your** installation. Appendix A is split between Volume 1 and Volume 2.

Appendix A (Volume 1) contains findings related to the service policy in effect and related to general system findings.

Appendix A (Volume 2) contains specific findings related to the analysis CPExpert performs if CPExpert detected that a service class did not meet its performance goal.

©Copyright 1994, Computer Management Sciences, Inc. Revised: October, 1997

Acknowledgements

Computer Management Sciences would like to acknowledge the following individuals. They played a significant role in the concept, design, development, testing, or documentation of CPExpert[1]:

John Ebner, Litton Computer Services

Alan Greenberg, Social Security Administration

Stan Meacham, MCI Corporation

Barry Merrill, Merrill Consultants

Philip Mugglestone, European Software Product Services, NV

Bryant Osborn, Litton Computer Services

John Peterson, MCI Corporation

Bernie Pierce, IBM Corporation

Fred Voth, JOSTENS Corporation

CPExpert would not exist as a product if it had not been for the personal inspiration, professional advice, technical knowledge, and encouragement from these individuals!

Additionally, Computer Management Sciences would like to acknowledge the following individuals. They played a significant role in the concept, design, development, and testing of the WLM Component of CPExpert:

Ed Berkel, IBM Corporation

David Cogar, DYNCORP

V. Lee Conyers, US Department of Transportation

Peter Enrico, IBM Corporation

Barry Merrill, Merrill Consultants

Bryant Osborn, US Department of Transportation

[1]The affiliation of the individuals is shown as of the time CPExpert was developed. Some of the individuals are no longer associated with the organizations shown.

Contents

Page

Changes ix

Section 1: Introduction

Chapter 1: IBM's Workload Manager 1-1

Chapter 2: The WLM Component of CPExpert 1-3

Chapter 3: Data Sources 1-5

Chapter 4: Performance Data Bases 1-6

Section 2: Specifying Guidance Variables

Chapter 1: Data Selection and Presentation Variables 2-2
Chapter 1.1: Specifying data selection guidance 2-3
Chapter 1.1.1: WLMDATES and WLMTIMES variables 2-3
Chapter 1.1.2: WLMDATEE and WLMTIMEE variables 2-3
Chapter 1.1.3: WLMDAT2S and WLMTIM2S variables 2-3
Chapter 1.1.4: WLMDAT2E and WLMTIM2E variables 2-4
Chapter 1.1.5: SHIFT variable 2-4
Chapter 1.1.6: N_PLEX variable 2-5
Chapter 1.1.7: SYSPLEX variable 2-5
Chapter 1.1.8: SYSTEM variable 2-6
Chapter 1.1.9: SYSTEMn variables 2-7
Chapter 1.1.10: SELECTSW, SELECTn, and PERIODn variables 2-7
Chapter 1.1.11: EXCLUDSW and EXCLUDEn variables 2-10
Chapter 1.2: Specifying data presentation guidance 2-11
Chapter 1.2.1: VERBOSE variable 2-11
Chapter 1.2.2: LISTRPT variable 2-12
Chapter 1.2.3: POLORDER variable 2-12
Chapter 1.2.4: WLMAST variable 2-13
Chapter 1.3: SAS Output Delivery System 2-13

Chapter 2: Analysis Guidance Variables 2-14
Chapter 2.1: AVGRESP variable 2-15
Chapter 2.2: CHKPLCY variable 2-15
Chapter 2.3: EXECSAMP variable 2-16
Chapter 2.4: HIGHCPU variable 2-17

©Copyright 1994, Computer Management Sciences, Inc.

Page

Chapter 2.5: INACTIVE variable 2-17
Chapter 2.6: MAXDUR variable 2-18
Chapter 2.7: MAXRESP variable 2-18
Chapter 2.8: MAXVEL variable 2-19
Chapter 2.9: MINSAMP variable 2-20
Chapter 2.10: MINTRANS variable 2-21
Chapter 2.11: OKPAGEIN variable 2-21
Chapter 2.12: PCTSERVC variable 2-22
Chapter 2.13: PERFINDX variable 2-23
Chapter 2.14: PHASE variable 2-24
Chapter 2.15: POLCHG variable 2-24
Chapter 2.16: SWAPDW variable 2-25
Chapter 2.17: SWAPTO variable 2-26
Chapter 2.18: Turning OFF WLM Component Rules 2-27
Chapter 2.19: WLMSIG variable 2-28

Chapter 3: XCF Performance Analysis Guidance Variables 2-29
Chapter 3.1: COUPLE variable 2-32
Chapter 3.2: EXCLASSW and EXCLASSn variables 2-32
Chapter 3.2.1: EXCLASSW variable 2-32
Chapter 3.2.2: EXCLASSn variable(s) 2-33
Chapter 3.3: PCTSML variable 2-33
Chapter 3.4: PCTBIG variable 2-35
Chapter 3.5: PCTREJ variable 2-35
Chapter 3.6: LOCKCONT variable 2-36
Chapter 3.7: FALSECNT variable 2-37
Chapter 3.8: SYNCSRV variable and LOCKSRV 2-38
Chapter 3.9: ASYNCSRV variable 2-39
Chapter 3.10: SYNCCHG variable 2-39
Chapter 3.11: Specifying guidance for specific structures 2-40

Chapter 4: System Logger Analysis Guidance Variables 2-42
Chapter 4.1: SMFTYP88 variable 2-45
Chapter 4.2: LGDSFULL 2-46
Chapter 4.3: LGSHIFTS 2-46
Chapter 4.4: PCTINTST 2-47
Chapter 4.5: PCTLOCT 2-48
Chapter 4.6: STDSHIGH 2-50
Chapter 4.7: STFULL90 2-52
Chapter 4.8: STRC2 2-53
Chapter 4.9: STRC3 2-54
Chapter 4.10: STRFULL 2-55
Chapter 4.11: Specifying guidance for specific log streams 2-56

Page

Section 3: Using the WLM Component

Chapter 1: Executing the WLMCPE Module 3-1
Step 1. Use TSO ISPF to change the "prefix" in the data set names 3-1
Step 2: Make any appropriate changes to the WLMGUIDE Module 3-3
Step 3: Review the output from the WLMCPE Module 3-3

Chapter 2: Checklist 3-4

Section 4: Analyzing the Workload Manager

Chapter 1: Workload Manager Concepts 4-2
Chapter 1.1: Service Definition 4-2
Chapter 1.2: Service Policies 4-2
Chapter 1.3: Workload Classification 4-3
Chapter 1.4: Service Classes 4-7
Chapter 1.4.1: Performance goals 4-9
Chapter 1.4.2: Performance goal importance 4-11
Chapter 1.4.3: Service class periods 4-11
Chapter 1.5: Report Classes 4-12
Chapter 1.6: Resource Groups 4-14
Chapter 1.7: Discretionary Goal Management 4-17
Chapter 1.8: Long-Term Storage Protection 4-19
Chapter 1.9 Long-Term CPU Protection 4-21
Chapter 1.10: Exemption from server management 4-23

Chapter 2: Subsystem Transactions 4-25
Chapter 2.1: Subsystem transaction service classes 4-25
Chapter 2.2: Subsystem interaction with Workload Manager 4-27

Chapter 3: Workload Manager Internal Logic 4-35
Chapter 3.1: Resource Use Information 4-35
Chapter 3.2: Domain Control Information 4-36
Chapter 3.3: Service Class Period Information 4-36
Chapter 3.4: Performance Index 4-40
Chapter 3.5: Policy Adjustment 4-42

Chapter 4: Workload Manager Performance Data 4-48

Chapter 5: RMF Data Analysis Considerations 4-51

©Copyright 1994, Computer Management Sciences, Inc. Revised: April, 2003

Page

Chapter 5.1: SRM Transactions 4-51
Chapter 5.2: RMF Transaction Information 4-53
Chapter 5.3: Operator Actions 4-55
Chapter 5.4: SRM Sampling of Subsystem State 4-56

Appendix A: Description of Rules

Exhibits

Page

2-1 Sample Display of prefix.CPEXPERT.USOURCE(WLMGUIDE) 2-2
2-2 Sample Display of prefix.CPEXPERT.USOURCE(WLMGUIDE) 2-11
2-3 Default values for XCF analysis in USOURCE(WLMGUIDE) 2-27
3-1 Job Control Language to execute the WLMCPE Module 3-2
4-1 Work Qualifiers available with each version of MVS 4-5
4-2 Example Service Classes 4-6
4-3 Relationship between Server and Served 4-25
4-4 Transaction Initialization 4-27
4-5 Subsystem Transaction States 4-28
4-6 Wait State Categories 4-29
4-7 Switched State Categories 4-31
4-8 Execution Phase States 4-32
4-9 End of Begin_to_end Phase 4-33
4-10 Performance Index Effect 4-40
4-11 Sample Ordering of Service Classes 4-44
4-12 Response Time Distribution, SMF Type 72 Records 4-48

Changes

CPExpert Release 13.1:

The main changes to the WLM Component for CPExpert Release 13.1 are to:

- Updates have been provided for neuMICS, to analyze additional problems or potential problems that occur with Service Definition specifications. The WLM Component analyzes such areas as:

  - Conflicts between Dynamic Alias Management specifications for Parallel Access Volume (PAV) and specifications for I/O Priority Management.

  - Conflicts between CPU Critical specifications for transaction service classes and their associated CICS or IMS region service classes.

  - Conflicts between job classes and workload executing in JES-managed initiators and WLM-managed initiators.

- With Release 13.1, the output from the WLM Component can be produced based on "level of impact" of each finding. Some CPExpert users with large sites were receiving <u>many</u> pages of output from the WLM Component. This large output partially was because CPExpert was analyzing data from many systems, with 15-minute RMF intervals, and for 24 hours a day. Since such a large amount of output is difficult to review, options have been added to allow a user to specify that the output should be available on a summary basis, for only important findings, or for all findings.

- Add the ability to select up to 10 systems individually for analysis. Until Release 131, a user had the options of analyzing data for all systems in the performance data based, analyzing data for a specific sysplex (in case the performance data base contained data for more than one sysplex), or analyzing data for a specific system in the performance data base. With Release 13.1, up to 10 systems can be individually selected for analysis.

- Enhance the options provided with the SAS Output Delivery System (ODS). With Release 13.1, users who exercise the SAS ODS feature for creating CPExpert output can optionally create the output as a PDF file, which can be emailed to other users. Additionally, users can optionally specify a STYLE feature for either HTML or PDF output, if they have a preferred STYLE for HTML or PDF output. The optional links that are available with the HTML have been revised; SAS at some user sites did not create the HTML output in the "standard" way, and the CPExpert code that inserted links into the HTML output did not work properly. I have revised the code to place the links into the output as the output is created, rather than attempting to place the links into the final HTML output created by SAS.

CPExpert Release 12.2:

The main changes to the WLM Component for CPExpert Release 12.2 are to:

- Update the Component to support z/OS Version 1 Release 4.
- Add Rule WLM031 to analyze conflicts between Dynamic Alias Management specifications for Parallel Access Volume (PAV) and specifications for I/O Priority Management.
- Add Rule WLM032 to analyze conflicts between CPU Critical specifications for transaction service classes and their associated CICS or IMS region service classes.
- Provide an introductory listing of significant analysis that cannot be done by CPExpert because (1) files are missing in the performance data base being evaluated or (2) analysis options have not directed the WLM Component to perform the analysis.
- Correct code based on errors reported by users.

CPExpert Release 12.1:

The main changes to the WLM Component for CPExpert Release 12.1 are to update the Component to support z/OS Version 1 Release 3.

CPExpert Release 11.2:

The main changes to the WLM Component for CPExpert Release 11.2 are to:

- Update the Component to support z/OS Version 1 Release 2.
- Add support for the SAS Output Delivery System (ODS) feature, to enable optional web access of CPExpert reports (this new option was suggested by **Harald Seifert** of HUK, Coburg, Germany).
- Add optional links in CPExpert reports (if the SAS ODS feature is invoked), that link rule output to CPExpert documentation for the rules produced.

CPExpert Release 11.1:

The main changes to the WLM Component for CPExpert Release 11.1 are to:

- Update the Component to support z/OS Version 1 Release 1.

- Added SYSOTHER analysis for MICS.
- Enhanced the data shaping code to improve execution performance.
- Created option to process all data in a performance data base with more than one sysplex. The new code optionally analyzes each sysplex in the performance data base.
- Enhanced code to analyzed function-shipped transactions, analyzing the transactions both on the originating and supporting system.
- Added analysis of IMS subsystem delay states for transaction service classes (applicable only with IMS Version 5 and subsequent versions of IMS).
- Added analysis of DB2 subsystem delay states for transaction service classes (applicable only with DB2 Version 6 and subsequent versions of DB2).
- Changed analysis of whether log stream interim storage was efficiently used to report on both log stream and the coupling facility structure involved.
- Eliminate report classes from server delay analysis (report classes can have server information reported in the SMF Type 72 Service Class Served Data Section, but all work units may not be servers).
- Correct code based on errors reported by users.

CPExpert Release 10.2:

The main changes to the WLM Component for CPExpert Release 10.2 are to:

- Update the Component to support OS/390 Version 2 Release 10.
- Correct code based on errors reported by users.

CPExpert Release 10.1:

The main changes to the WLM Component for CPExpert Release 10.1 are to:

- Update the Component to support OS/390 Version 2 Release 9.
- **WLM-managed initiator analysis.** The following new rules have been added to analyze your WLM-managed initiator/job class assignment scheme:

- A rule to analyze whether JES-managed and WLM-managed job classes conflict by being assigned to the same service class period, with a resulting lack of efficient management by the Workload Manager.

- A rule to analyze whether WLM-managed job class were assigned to multiple service classes, with a resulting lack of efficient management by the Workload Manager.

- A rule to analyze whether job(s) or job class(es) might not be suitable for assignment to WLM-managed initiators, due to the tendency of the execution queue delay to dominate total execution time.

- **System logger analysis**. The following new rules have been added to the WLM Component to analyze performance problems with your system logger:

  - A rule to analyze when the log stream coupling facility structure was full.

  - A rule to analyze when the log stream staging data set was full.

  - A rule to analyze when the log stream structure offloads occurred because the coupling facility structure was 90% full.

  - A rule to analyze when interim storage was not efficiently used for log stream.

  - A rule to analyze when local storage buffers were not efficiently used, for a DASD-only log stream.

  - A rule to analyze when the DASD staging data set high threshold was reached for a log stream.

  - A rule to analyze when frequent log stream DASD-shifts (with concurrent unnecessary overhead) occurred.

  - A rule to analyze whether a log stream caused a coupling facility structure to reach the high threshold (HIGHOFFLOAD).

  - A rule to analyze situations when a log stream consumed most of a coupling facility structure's resources.

CPExpert Release 9.2:

The main changes to the WLM Component for CPExpert Release 9.2 are to:

- Update the Component to support OS/390 Version 2 Release 8.

- Provide the ability to specify analysis guidance for individual structures in a coupling facility. The guidance variables for structures normally are globally applied during CPExpert's analysis of structure performance. These global guidance variables might not be applicable to some structures, however. With this change, guidance can **optionally** be applied to specific structures.

- Add an option to suppress the analysis and findings of particular rules. This feature typically is desired by users who (1) have an overall disagreement with the finding, (2) are unable to make a suggested change, or (3) decide that a particular finding is inapplicable to a their environment.

- Correct code to account for errors discovered in IBM's SMF documentation.

- Correct documentation and code based on errors reported by users.

CPExpert Release 9.1:

The main changes to the WLM Component for CPExpert Release 9.1 are to:

- Update the Component to support OS/390 Version 2 Release 7.

- Add a new rule to analyze situation when Execution Phase samples for CICS transaction service class data did not exist in SMF Type 72 records (Work Manager/Resource manager section).

- Add a new rule to advise that the response performance goal may be too large.

- Add a new rule to advise that a service class was capped for discretionary goal management.

- Add a new rule to advise that non-DASD I/O activity or delay was significant.

- Add a new rule to advise that non-paging DASD I/O activity caused significant delays.

- Correct code to account for errors discovered in IBM's SMF documentation.

- Correct documentation and code based on errors reported by users.

CPExpert Release 8.2:

The main changes to the WLM Component for CPExpert Release 8.2 are to:

- Update the Component to support OS/390 Version 2 Release 6.
- Add analysis to support discretionary goal management introduced with OS/390 Version 2 Release 6.
- Enhance the code to remove SAS work files when no longer needed. This feature was required for users with extremely large performance data bases. These users experienced significant increases in the amount of SAS work space required for CPExpert to analyze data. The amount of SAS work space was significantly decreased by deleting CPExpert work files when they were no longer required by the logic.
- Add the SYSPLEX variable to all data selection criteria. This data selection option is required for large MICS sites that have data from more than one sysplex in their performance data base.
- Add the MICSLEV guidance variable to allow users to specify their change level of MICS. This variable is required since MICS does not retain information describing the installed MICS change level.
- Correct MICS variables with OS/390 Version 1 Release 3.
- Add code to account for changes in MXG 16.04.
- Correct MICS coding errors with MICS Change RMF6380.
- Change all CPExpert coding references from &PDBLIB to &WLMLIB. This change was required for users who retained Goal Mode data in a separate SAS library.
- Add the operating system level and MXG version (if MXG performance data base) to the CPExpert output listing information.
- Modify the local page data set analysis to perform some test regardless of whether a performance problem existed with service class periods.
- Add a user option to control whether CPExpert more dramatically separates analysis results for service class periods that missed their performance goal.
- Correct code based on errors reported by users.

CPExpert Release 8.1:

The main changes to the WLM Component for CPExpert Release 8.1 are:

- Update the Component to support OS/390 Version 2 Release 5.

- Add support for MVS Goal Mode information residing in a CA-MICS performance data base. Thanks to **Gerd Krohm** (Westdeutsche Landesbank, Düsseldorf, GERMANY) and **Bryant Osborn** (NationsBank, Richmond, Virginia) for providing CA-MICS test data so the WLM Component could be enhanced to support CA-MICS.

- Add code that will recognize and discard certain erroneous data in SMF/RMF records.

- Correct code based on errors reported by users.

Section 1: Introduction

This section provides a brief overview of IBM's Workload Manager, provides an overview of the WLM Component of CPExpert, describes the sources of data used by the WLM Component to analyze system performance, and describes the performance data bases that the WLM Component can use.

Chapter 1: IBM's Workload Manager

IBM's Workload Manager is a radical departure from earlier system management controls provided with MVS.

- With earlier versions of MVS, users provided MVS with detailed specifications about how MVS should process work.

  - Users classified workloads using the IEAICSxx member of SYS1.PARMLIB, and assigned the resulting workload classes to performance groups.

  - Domains and performance groups were defined in the IEAIPSxx member of SYS1.PARMLIB, and the domain and performance group specifications provided detailed guidance to MVS about how to process the work associated with the domains and performance groups.

  - System parameters contained in the IEAOPTxx member of SYS1.PARMLIB provided specifications to MVS at an overall system view. These specifications guided (or in some cases, actually controlled) MVS in how work was to be processed by the system. The specifications provided system-wide multiprogramming adjustment controls, system-wide expanded storage guidance (MVS/ESA SP4.2 introduced expanded storage guidance at the domain level), system-wide logical swap guidance, etc.

  Systems personnel were required to understand the detailed internal logic of the System Resources Manager (SRM) part of MVS and how the system parameters effected that logic, in order to provide proper specifications in the SYS1.PARMLIB members. The MVS *Initialization and Tuning Guide* and *Initialization and Tuning Reference* documents described the logic and parameters, but the descriptions often were incomplete or misleading.

  The SRM is complex, the concepts in the MVS *Initialization and Tuning Guide* and *Initialization and Tuning Reference* are difficult to appreciate, and SRM parameters often have unexpected results. Papers are regularly presented at professional conferences to describe newly-discovered performance problems or solutions resulting from experiences with the System Resources Manager[1]. Further, many organizations

[1]In fact, the whole concept of CPExpert arose because of a perceived need to have a tool to help analyze the performance of MVS.

did not even attempt to "tune" the SRM parameters because they did not understand the concepts. Many organizations simply used the IBM default values (even though IBM documents explicitly stated that the defaults should be changed).

- The Workload Manager attempts to solve theses problem by removing the requirement that users provide detailed guidance to MVS on how MVS should process work. With the Workload Manager, users specify **performance goals**. The Workload Manager interprets the performance goals, and the Workload Manager tells MVS how to process work to meet the performance goals.

  The major difference between earlier versions of MVS and the Workload Manager is that users do not provide the Workload Manager with detailed specifications on how to process the work assigned to a service class. Users describe **service** requirements, and the Workload Manager adjusts system resources to meet the service requirements.

IBM has provided a wealth of information about system performance and constraints to performance with the Workload Manager. The primary source of performance information about the performance of service classes is contained in SMF Type 72 (Subtype 3) records[2]. The Type 72 records contain detailed information about performance goals, information which can be analyzed to determine whether the performance goals were met, and information showing performance constraints.

The Workload Manager is a new concept and new software. Consequently, there will be an industry-wide "learning curve" as installations gain experience using the Workload Manager to meet performance goals. The WLM Component of CPExpert can help reduce the learning curve. Additionally, the WLM Component will be improved as additional experience is gained by our users and by the industry as a whole.

Further, it is likely that some changes will be required in both the Workload Manager concept and the Workload Manager software to accommodate unique situations. The WLM Component of CPExpert will be modified as necessary to respond to these changes.

[2]The SMF Type 99 records contain extremely detailed information about the internal logic and decisions of the Workload Manager. The Workload Manager creates a "trace" of its logic as it examines service classes, makes resource adjustment decisions, makes policy adjustment decisions, makes working set management decisions, etc. The Type 99 records are written every 10 seconds, and may be extremely large. IBM strongly recommends that Type 99 records be collected for only a short time in response to specific needs. Since the Type 99 records are not normally collected, CPExpert does not consider them to be a **primary** source of performance data. In contrast, the Type 72 records are routinely collected at virtually every site.

©Copyright 1994, Computer Management Sciences, Inc. Revised: June, 2001

Chapter 2: The WLM Component of CPExpert

The WLM Component consists of numerous modules, working together to (1) detect intervals in which service classes did not achieve performance goals, (2) shape system performance and utilization data for detailed analysis by other modules, (3) evaluate the data to assess reasons service classes did not achieve performance goals, and (4) report the results from the evaluation. Additionally, the WLM Component contains modules to analyze the Workload Manager service definition to detect potential problems with workload classification or service class definitions. These modules are loaded and controlled by the central WLM Component of CPExpert (titled WLMCPE).

- **Detect missed performance goals**. The WLM Component analyzes SMF Type 72 (Subtype 3) data to detect measurement intervals when service classes did not achieve response or execution velocity performance goals.

- **Shape Data**. The majority of "processing" in the WLM Component is accomplished by numerous modules whose function is to combine, sort, summarize, correlate, and prepare the data for analysis. These modules consolidate relevant data elements from SMF Type 70, Type 71, Type 72, Type 74, and Type 75 into one record for each service class that missed its performance goal during an entire RMF measurement interval. This record contains all information necessary to perform a comprehensive analysis of the system performance areas addressed by the WLM Component.

  Additionally, data from SMF Type 30 records may be shaped and consolidated if the DASD Component of CPExpert is licensed and the appropriate modification has been made to MXG or MICS code. Please refer to the DASD Component User Manual for details.

- **Evaluate Data**. The evaluation of the data is accomplished by rules whose purpose is to (1) evaluate the Workload Manager service policy, (2) identify problems or potential problems with the overall system, and (3) isolate reasons a service class did not meet its goal.

- **Report Results**. The WLM Component reports the results from the evaluation, in either "verbose" or "summary" reporting. The verbose reporting method presents a variety of data leading to a particular finding. The summary reporting method produces only a time-related graphical overview of the findings.

- **Maintain Historical Data**. The WLM Component optionally maintains a historical file of all rule results. This file is available for subsequent analysis by users who may wish to trend the frequency of different findings by the WLM Component.

©Copyright 1994, Computer Management Sciences, Inc.

Each rule that results in a finding is described in Appendix A of this document. The description summarizes the rule, lists predecessor rules, discusses the rationale for the finding, suggests action, and lists references related to the finding.

- The summary of the rule presents a short description of the finding.

- The predecessor rules are listed so you can follow the line of reasoning leading to a particular rule being executed.

- The discussion of the finding describes as much as necessary of the operation of the computer system (the Workload Manager, the SRM, MVS, the system hardware, etc.) as it relates to the particular rule. The purpose of the discussion is to explain the reasoning behind the rule, and what causes the rule to be produced. If appropriate, the discussion might refer you to related rules documented in the WLM Component User Manual.

- The suggestions list possible actions that should be considered based on the findings. In many cases, multiple possible actions are listed. You must determine which actions should be taken (this determination is based upon the suitability of the actions to your own environment, the financial implications of the action, and the "political" acceptability of the action.)

- References are listed with many rule descriptions. The references are provided so you can verify the analysis and suggestions.

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 3: Data Sources

CPExpert analyzes the performance of your system operating under the Workload Manager based upon data from two sources:

- **Guidance information**. Guidance information is contained in prefix.CPEXPERT.USOURCE(WLMGUIDE). The variables in the WLMGUIDE module can be viewed as "data selection and presentation" variables and "analysis control" variables.

  - The data selection and presentation variables allow you to select particular time intervals to be analyzed, and allow you to specify how the results from the analysis are to be presented.

  - The analysis control variables allow you to control the analysis the WLM Component will perform. These variables establish threshold values which are used by the WLM Component to assess system performance

    The defaults provided for the analysis control variables will be appropriate for many environments. However, if your environment is unique, you can alter the analysis by changing the analysis control variables.

    You should not hesitate to alter the guidance variables to meet your requirements. Some of the default values are deliberately set to cause rules to be produced initially. The purpose of this is to call your attention to the performance implications of certain decisions you may have made. Those decisions may be appropriate for your environment, even though CPExpert may "flag" them as potential problems.

- **RMF information**. CPExpert does most of its analysis based upon the information acquired by RMF and recorded into SMF Type 70(series) records. The primary information is contained in the SMF Type 72 records. IBM has included virtually everything necessary to analyze constraints to improved performance (from the view of the Workload Manager) in the Type 72 records: the service policy definition, the performance of service classes, delays to service classes, and performance and delays to subsystem transactions (if CICS Version 4 or IMS Version 5 are installed).

  If installations do not operate RMF, they often operate a product that produces similar information. In particular, CPExpert can analyze performance using the information provided the Capacity Management Facilities product from Boole & Babbage, Inc. if the data is available in a MXG or MICS performance data base.

©Copyright 1994, Computer Management Sciences, Inc. Revised: June, 2001

Chapter 4: Performance Data Bases

The "raw" RMF data contained in the SMF Type 70(series) records must be translated into SAS format and placed into a SAS-based performance data base before CPExpert can use the information. There are three ways in which these records normally are placed into a performance data base:

- **MXG performance data base**. The performance data base can be created by Merrill's Expanded Guide (MXG) software. MXG is provided by Merrill Consultants, Dallas, Texas. MXG provides a low-cost mechanism by which installations can create and maintain a performance data base.

- **MICS performance data base**. The performance data base can be created by the MVS Integrated Control System (MICS). MICS is provided by Legent Corporation, Vienna, Virginia. The MICS SMF (or Base) Component creates and maintains a comprehensive performance data base from SMF/RMF information.

- **Locally-created performance data base**. The performance data base can be one created locally from "raw" SMF Type 70(series) records. At present, no user of CPExpert has applied CPExpert to a locally-created performance data base. Please contact Computer Management Sciences for guidance if you wish to apply CPExpert against a locally-created performance data base.

The flexibility to use any of the above performance data bases is due to the fact that CPExpert is implemented in the SAS language. SAS provides a powerful macro facility, both with respect to macro coding and with respect to macro variable names.

CPExpert uses SAS macro variable names when referring to an element of information in the SMF records. CPExpert uses the SAS "%LET" statements to define the macro variables as MXG variable names, MICS variable names, or SMF basic variable names. These "%LET" statements are contained in unique variable definition modules for MXG, MICS, or raw SMF.

Thus, the same CPExpert software can be executed against any of the three types of performance data bases, by only invoking the proper definition module. The SAS %LET statements in the definition module automatically cause CPExpert to refer to the proper MXG, MICS, or SMF data elements.

Some important variables are not naturally available in the SMF Type 70(series) information, but are computed or derived by MXG and MICS. For these variables, CPExpert arbitrarily chose the MXG variable name or chose an "artificial" SMF name [e.g., "SMF72CPU" represents the combined R723CCPU and R723CSRB times for service class periods represented in SMF Type 72 (Subtype 3) records].

©Copyright 1994, Computer Management Sciences, Inc. Revised: June, 2001

Section 2: Specifying Guidance Variables

The WLM Component must evaluate performance constraints of systems operating under Goal Mode with a wide variety of configuration options, use data from several different sources, and evaluate the constraints to improved performance from the perspective of many different management objectives. Guidance variables are provided to the WLM Component to allow it to respond to the different configurations, different data sources, and different management objectives.

The prefix.CPEXPERT.USOURCE(WLMGUIDE) PDS member contains variables to establish the overall guidance for the WLM Component. You modify the variables in the WLMGUIDE member whenever you wish to change the guidance to CPExpert. This chapter describes these variables, how the variables are used, and how the variables are altered.

The variables in the WLMGUIDE module can be viewed as "data selection and presentation" variables and "analysis control" variables. These two types of control variables are discussed separately.

- The *data selection and presentation variables* allow you to select particular time intervals to be analyzed, allow you to select or exclude specific service classes, and allow you to specify how the results from the analysis are to be presented.

- The *analysis control variables* allow you to control the analysis the WLM Component will perform. The defaults provided for the analysis control variables will be appropriate for many environments. However, if your environment is unique, you can alter the analysis by changing the analysis control variables.

You should not hesitate to alter the guidance variables to meet your requirements. Some of the default values are deliberately set to cause rules to be produced initially. The purpose of this is to call your attention to the performance implications of certain decisions you may have made. Those decisions may be appropriate for your environment, even though CPExpert may "flag" them as potential problems.

Please do not allow CPExpert to perform analysis or produce reports which are meaningless in your environment. If the analysis and reports produced by CPExpert do not meet your needs, alter the guidance to CPExpert. If the guidance is insufficient, please call Computer Management Sciences at (703) 922-7027 (or e-mail **Don_Deese@cpexpert.com**) so we can make changes to improve CPExpert for you!

Chapter 1: Data Selection and Presentation Variables

The data selection and presentation variables allow you to select particular time intervals to be analyzed, and allow you to specify how the results from the analysis are to be presented. This chapter describes these variables.

Exhibit 2-1 illustrates the portion of CPEXPERT.USOURCE(WLMGUIDE) that contains the data selection and presentation variables.

```
**** DATA SELECTION AND PRESENTATION VARIABLES                                ;
%LET WLMDATES       =01FEB1994   ; * START DATE FOR DATA ANALYSIS            ;
%LET WLMTIMES       =08:00:00    ; * START TIME FOR DATA ANALYSIS            ;
%LET WLMDATEE       =31DEC9999   ; * END DATE FOR DATA ANALYSIS              ;
%LET WLMTIMEE       =17:59:59    ; * END TIME FOR DATA ANALYSIS              ;
%LET WLMDAT2S       =0           ; * DEFAULT SECOND SELECTION DATE - START   ;
%LET WLMTIM2S       =0           ; * DEFAULT SECOND SELECTION TIME - START   ;
%LET WLMDAT2E       =0           ; * DEFAULT SECOND SELECTION DATE - END     ;
%LET WLMTIM2E       =0           ; * DEFAULT SECOND SELECTION TIME - END     ;
%LET SHIFT          =N           ; * START AND END TIMES DO NOT APPLY TO SHIFT;
%LET N_PLEX         =N           ; * VARIABLE TO CONTROL ANALYSIS OF SYSPLEX  ;
%LET SYSPLEX        =*ALL        ; * SPECIFY SYSPLEX TO PROCESS (*ALL = ALL)  ;
%LET SYSTEM         =*ALL        ; * SPECIFY SYSTEM TO PROCESS (*ALL = ALL)   ;
%LET SYSTEMn        =system      ; * PROCESS SYSTEMn (n = 1-9)               ;
%LET SELECTSW       =N           ; * DO NOT SELECT SPECIFIC SERV CLASSES     ;
%LET SELECT1        =TSO         ; * SAMPLE:  ANALYZE TSO SERVICE CLASS      ;
%LET PERIOD1        =1           ; * SAMPLE: ANALYZE TSO PERIOD 1            ;
%LET EXCLUDSW       =N           ; * DO NOT EXCLUDE SPECIFIC SERVICE CLASSES ;
%LET EXCLUDE1       =BATCHLOW    ; * SAMPLE: EXCLUDE BATCHLOW SERVICE CLASS  ;
%LET VERBOSE        =V           ; * RESULTS:  VERBOSE/CONCISE/SUMMARY       ;
%LET LISTRPT        =N           ; * DO NOT LIST REPORT CLASSES              ;
%LET POLORDER       =ALPHA       ; * ORDER TO LIST POLICY (ALPHA OR IMPORT)  ;
%LET WLMAST         =Y           ; * VARIABLE TO CONTROL WLM RULE LISTING    ;
%LET SASODS         = N          ; * CONTROLS WHETHER SAS ODS IS USED        ;
%LET PATH           =            ; * PATH FOR ODS OUTPUT                    ;
%LET FRAME          = WLMFRAME   ; * GENERIC ODS FRAME NAME                 ;
%LET CONTENTS       = WLMDCONT   ; * GENERIC ODS CONTENTS NAME              ;
%LET BODY           = WLMBODY    ; * GENERIC ODS BODY NAME                  ;
%LET STYLE          =            ; * ODS HTML STYLE OPTION                  ;
%LET PDFODS         = N          ; * CONTROLS WHETHER SAS PDF IS USED       ;
%LET LINKPDF        =            ; * LINK TO CPEXPERT DOCUMENTATION         ;
%LET URL            = N          ; * CONTROLS .HTM IN SAS ODS FRAME OUTPUT  ;
```

SAMPLE DISPLAY OF prefix.CPEXPERT.USOURCE(WLMGUIDE) MODULE

EXHIBIT 2-1

Chapter 1.1: Specifying data selection guidance

CPExpert allows you to select specific measurement intervals for analysis. This facility is controlled by the WLMDATES, WLMTIMES, WLMDATEE, WLMTIMEE, WLMDAT2S, WLMTIM2S, WLMDAT2E, and WLMTIM2E variables. This chapter describes these variables.

Chapter 1.1.1: WLMDATES and WLMTIMES variables

The WLMDATES and WLMTIMES variables are required (although the defaults values may be used to analyze all data through 1999). These specify the start date and start time, respectively, for the interval of SMF data the WLM Component is to analyze. These variables (in conjunction with the WLMDATEE and WLMTIMEE variables) allow you to select specific periods of data to analyze. For example, to specify that data selection should start at 08:00:00 on March 8, 1995, specify:

```
%LET WLMDATES = 08MAR1995;      * START DATE FOR DATA ANALYSIS;

%LET WLMTIMES = 08:00:00;       * START TIME FOR DATA ANALYSIS;
```

Chapter 1.1.2: WLMDATEE and WLMTIMEE variables

The WLMDATEE and WLMTIMEE variables are required (although the defaults values may be used to analyze all data through 1999). These variables specify the end date and end time, respectively, for the interval of SMF data the WLM Component is to analyze. For example, to specify that data selection should end at 17:00:00 on March 8, 1995, specify:

```
%LET WLMDATEE = 08MAR1995;      * END DATE FOR DATA ANALYSIS;

%LET WLMTIMEE = 17:00:00;       * END TIME FOR DATA ANALYSIS;
```

Chapter 1.1.3: WLMDAT2S and WLMTIM2S variables

The WLMDAT2S and WLMTIM2S variables are optional. These variables specify the start date and start time, respectively, for a second interval of SMF data the WLM Component is to analyze. These variables (in conjunction with the optional WLMDAT2E and WLMTIM2E variables) allow you to select a second period of data to analyze, in addition to the period specified by the WLMDATES/WLMTIMES and

WLMDATEE/WLMTIMEE selection variables. For example, to specify that a second period of data selection should start at 20:00:00 on March 8, 1995, specify:

```
%LET WLMDAT2S = 08MAR1995;      * START DATE FOR DATA ANALYSIS;

%LET WLMTIM2S = 20:00:00;       * START TIME FOR DATA ANALYSIS;
```

Chapter 1.1.4: WLMDAT2E and WLMTIM2E variables

The WLMDAT2E and WLMTIM2E variables are optional. These variables specify the end date and end time, respectively, for a second interval of SMF data the WLM Component is to analyze. These variables (in conjunction with the optional WLMDAT2S and WLMTIM2S variables) allow you to select a second period of data to analyze, in addition to the period specified by the WLMDATES/WLMTIMES and WLMDATEE/WLMTIMEE selection variables. For example, to specify that a second period of data selection should end at 22:00:00 on March 8, 1995, specify:

```
%LET WLMDAT2E = 08MAR1995;      * END DATE FOR DATA ANALYSIS;

%LET WLMTIM2E = 22:00:00;       * END TIME FOR DATA ANALYSIS;
```

Chapter 1.1.5: SHIFT variable

The SHIFT variable is used with the WLMDATES, WLMTIMES, WLMDATEE, and WLMTIMEE variables (and the optional variables to select a second period for analysis). The SHIFT variable allows you indicate how the time-selection variables should be used.

- If the SHIFT variable is "N", the time-selection will be based upon the **absolute** start and end dates/times specified. For example, if you wish CPExpert to process **all** data during a week, the start date and start time would be specified as the beginning of the week, and the end date and end time would be specified as the end of the week. You would specify "%LET SHIFT = N;" to process each 24-hour day.

- If the SHIFT variable is "Y", the time-selection will be based upon the start and end dates, and the start and end times within each selected date. In the example shown above, perhaps you wished to process only the daily shift beginning at 08:00:00 and ending at 17:00:00. You would specify "%LET SHIFT = Y;" to process only the identified shift data, during the selected dates.

©Copyright 1996, Computer Management Sciences, Inc.

Chapter 1.1.6: N_PLEX variable

The N_PLEX variable is used to specify whether the performance data base contains information from more than one sysplex.

Under normal circumstances, the WLM Component analyzes all data in a performance data base. However, some large sites have a performance data base containing data from more than one sysplex. For these sites, CPExpert must analyze the data on a sysplex-by-sysplex basis[1]. Analyzing data on a sysplex-by-sysplex basis requires substantial additional processing resources. It is not reasonable that these processing resources be expended if the performance data base contains only one sysplex. Nor is it reasonable to expend the processing resources if the CPExpert user wishes to analyze only a specific sysplex (for example, a user might not wish to analyze data relating to a test sysplex).

In order to minimize processing resources required to execute CPExpert, the N_PLEX guidance variable was introduced. The N_PLEX guidance variable simply specifies whether the performance data base contains data relating to more than one sysplex, and data from each sysplex should be analyzed.

The default specification for the N_PLEX guidance variable is **%LET N_PLEX=N;** indicating that the performance data base contains data for only one sysples. If the perfomance data base contains data from more than one sysplex and you wish to analyze each, you **must** either specify **%LET N_PLEX-Y;** in USOURCE(WLMGUIDE), or select a specific sysplex to analyze.

Chapter 1.1.7: SYSPLEX variable

The SYSPLEX variable is used to specify whether data from each sysplex in the performance data base should be evaluated, or whether CPExpert should select data for a specific sysplex to be evaluated.

Some users have data from more than one sysplex in their performance data base. For many of these users, or for users who have data for a single sysplex represented in their performance data base, the default "*ALL" will be appropriate. No change of the SYSPLEX variable would be required for these users (although the N_PLEX variable would be required, as described above).

However, some users who have data from more than one sysplex may wish to evaluate only a single sysplex with the parameters specified in this member of WLMGUIDE. This evaluation can be accomplished by changing the SYSPLEX variable to specify the sysplex

[1] This is necessary because some analysis (for example, computing a sysplex-wide Performance Index) requires that CPExpert process all data in the performance data base that relates to a specific sysplex, regardless of whether the user has specified that analysis be performed of only a specific system.

©Copyright 1996, Computer Management Sciences, Inc.

to be evaluated. The following example shows how to specify that data only from PRODPLEX should be evaluated:

```
%LET SYSPLEX = PRODPLEX ; * PROCESS DATA ONLY FROM PRODPLEX;
```

Chapter 1.1.8: SYSTEM variable

The SYSTEM variable is used to specify whether all systems in the performance data base should be evaluated, or to select a specific system identification to be evaluated. The default specification (**%LET SYSTEM=*ALL;)** tells CPExpert to analyze all systems encountered in the performance data base.

Some users have data from multiple systems in their performance data base. For many of these users, or for users who have data for a single system represented in their performance data base, the default "*ALL" will be appropriate. No change of the SYSTEM variable would be required for these users.

However, some users who have data from multiple systems may wish to evaluate only a single system with the parameters specified in this member of WLMGUIDE. For example, they might be temporarily interested in evaluating the performance of only an "important" system (such as a major production system) and not be interested in evaluating the performance of other systems with data in the performance data base. This evaluation can be accomplished by changing the SYSTEM variable to specify the system identification to be evaluated. For example, to specify that only data from SYS1 should be evaluated, specify:

```
%LET SYSTEM = SYS1 ;          * PROCESS ONLY DATA FROM SYS1;
```

In another situation, a CPExpert user might wish to evaluate different systems with different WLMGUIDE parameters. These different evaluations can be accomplished by different executions of the WLM Component. For each execution of the WLM Component, the USOURCE DD statement would be changed to reference different USOURCE libraries. Each USOURCE library would contain guidance members with appropriate guidance variables. The SYSTEM variable for each WLMGUIDE guidance member would specify the system identification to which the guidance applied.

Chapter 1.1.9: SYSTEMn variables

The SYSTEMn variables are used to select multiple systems to be evaluated.

As described in the SYSTEM guidance variable discussion above, some sites have data from multiple systems in their performance data base. These sites can process data from all systems by specifying %LET SYSTEM=ALL; in USOURCE(CICGUIDE), or can select a specific system to process by specifying %LET SYSTEM=system; in USOURCE(CICGUIDE), where “system” is the system identification of the system to be processed.

Some sites have data from multiple systems in their performance data base and do not want to process all systems, but do wish to process more than one system. For example, some systems might be production systems and some might be test systems. For these sites, the **SYSTEMn** guidance variable can be used to select more than one specific system to analyze.

The SYSTEM guidance variable can be used to seslect only one system to analyze, and the SYSTEMn guidance variable(s) can be used to select up to 9 additional systems to analyze. For example, if you wish to analyze data from four systems (named SYSA, SYSB, SYSC, AND SYSX) in a single execution of the WLM Component, specify:

```
%LET SYSTEM  = SYSA ; * PROCESS DATA FROM SYSA;
%LET SYSTEM1 = SYSB ; * PROCESS DATA FROM SYSB;
%LET SYSTEM2 = SYSC ; * PROCESS DATA FROM SYSC;
%LET SYSTEM3 = SYSX ; * PROCESS DATA FROM SYSX;
```

Chapter 1.1.10: SELECTSW, SELECTn, and PERIODn variables

You may wish to select specific service classes for analysis by the WLM Component. For example, you may wish to examine only your most important service classes.

Selecting service classes for analysis is optional; you do not have to select specific service classes for analysis. If you do not select specific service classes for analysis, CPExpert will analyze every service class encountered in SMF Type 72 data (optionally excluding service classes if the EXCLUDE option is exercised).

If you **do** select specific service classes for analysis, CPExpert normally will report on only those service classes. However, if you select a service class "served" by another service class, CPExpert will examine the "server" service class if the "served" service class does not achieve its goal.

©Copyright 1996, Computer Management Sciences, Inc.

- For example, suppose you have CICS/ESA 4.1 installed and you define a service class for particular CICS transactions. The CICS *region* would be associated with one service class (e.g., CICSRGN) and the particular CICS *transactions* would associated with other service classes (e.g., CICUSERA,CICUSERB, etc.). The subsystem work manager (in this case, CICS/ESA 4.1) would track performance of the transactions executing in the CICUSERA and CICUSERB Service Classes.

- You might SELECT the CICSTRAN Service Class containing the specific CICS transactions and direct CPExpert to analyze only that Service Class.

- Suppose CPExpert detects that the CICSTRAN Service Class did not achieve its response goal. CPExpert would analyze CICSTRAN to identify causes of poor performance. However, CPExpert also would analyze CICSRGN to identify causes of poor performance, since CICSRGN is the "server" Service Class.

- The Workload Manager normally does not allocate resources to "served" service classes (see Section 4 for exceptions), so no resources would be allocated to CICSTRAN. Rather, the Workload Manager would allocate resources to CICSRGN. Thus, CPExpert must analyze CICSRGN, even though CICSRGN was not a "selected" service class in the SELECT specifications.

If you wish to select service classes for analysis, you must (1) use the **SELECTSW** guidance variable, (2) define the service classes to select, and (3) optionally define the periods to select.

SELECTSW variable

The SELECTSW guidance variable is used to tell CPExpert whether you wish to select specific service classes for analysis. The SELECTSW variable acts as a "switch" to control whether CPExpert invokes the select logic to select service classes for analysis. The point of having a "switch" variable is that some installations may wish to regularly select specific service classes for analysis, but periodically analyze all service classes.

If the SELECTSW variable is **N**, CPExpert will not select any specific service classes for analysis, but will analyze all service classes encountered. If the SELECTSW variable is **Y**, CPExpert will select only the service classes specified by the SELECTn guidance variable(s).

Note that the SELECT logic applies only to service classes - not to report classes. CPExpert does not analyze report classes, as insufficient information is available with report classes.

©Copyright 1996, Computer Management Sciences, Inc.

SELECTn variables

You may select up to 9 service classes for analysis by coding the name of the service classes to select in the **%LET SELECTn** variables. For example, to select only the TSO Service Class, specify:

```
%LET SELECT1 = TSO;                    * SELECT TSO SERVICE CLASS;
```

It is not necessary that the SELECTn variables be specified in numerical order, nor is it necessary that all numbers be specified. For example, you can safely specify:

```
%LET SELECT6 = TSO;                    * SELECT TSO SERVICE CLASS;
%LET SELECT3 = CICSTRAN                * SELECT CICS SERVICE CLASS;
```

This feature is useful if you wish to temporarily "comment out" certain selections.

PERIODn variables

You may wish to SELECT particular service class periods. For example, you may be interested only in the performance of TSO Period 1 and TSO Period 2. You may wish that CPExpert not analyze TSO Period 3 or lower periods.

The **optional** PERIODn variable allows you to select specific service class periods for analysis. If you do **not** specify the PERIODn variable for a SELECTed service class, CPExpert will analyze **all** periods with the service class. For example, if you wish CPExpert to analyze **all** periods of the TSO Service Class, you may specify:

```
%LET SELECT4 = TSO;            * SELECT TSO SERVICE CLASS;
```

With this specification, CPExpert will analyze all periods in the TSO Service Class.

If you wish CPExpert to analyze only TSO Period 1 and TSO Period 2, you could specify:

```
%LET SELECT4=TSO;          * SELECT TSO SERVICE CLASS;
%LET PERIOD4=1;            * SELECT TSO PERIOD 1;
%LET SELECT5=TSO;          * SELECT TSO SERVICE CLASS;
%LET PERIOD5=2;            * SELECT TSO PERIOD 2;
```

With the above specification, CPExpert will analyze only TSO Period 1 and Period 2.

Note that you must completely identify the service class periods (that is, you must SELECT both the service class and the period). As shown above, the SELECT4 and PERIOD4 guidance variables apply to TSO Period 1, while the SELECT5 and PERIOD5 guidance variables apply to TSO Period 2. There must be a one-to-one correspondence between the numerical values of the SELECT and PERIOD guidance variables.

Chapter 1.1.11: EXCLUDSW and EXCLUDEn variables

For a variety of reasons, you may wish to exclude certain service classes from analysis. The most common reason is that CPExpert repeatedly identifies problems with a particular service class, but (1) you do not wish to make changes to correct the problems, (2) you are unable to make changes (because of application requirements or political realities), or (3) the WLM Component analysis is "flawed" because of data problems or data averaging. Whatever the reason, you may wish to exclude certain service classes from analysis.

Excluding service classes from analysis is optional; you do not have to exclude any service classes from analysis. If you do **not** exclude service classes from analysis, CPExpert will analyze every service class encountered in SMF Type 72 data (unless you have used the SELECT option to select specific service classes).

If you wish to exclude service classes from analysis, you must (1) use the **EXCLUDSW** guidance variable and (2) define the service classes to exclude.

EXCLUDSW variable

The EXCLUDSW guidance variable is used to tell CPExpert whether you wish to exclude service classes from analysis. The EXCLUDSW variable acts as a "switch" to control whether CPExpert excludes service classes from analysis. The point of having a "switch" variable is that some installations may wish to regularly exclude service classes from analysis, but periodically analyze all service classes.

If the EXCLUDSW variable is **N**, CPExpert will not exclude any service classes from analysis (although CPExpert may **select** only certain service classes if the SELECT options are used). If the EXCLUDSW variable is **Y**, CPExpert will exclude all service classes specified by the EXCLUDEn guidance variable(s).

Note that the EXCLUDE logic applies only to service classes - not to report classes. CPExpert does not analyze report classes, as insufficient information is available with report classes.

EXCLUDEn variable(s)

You may exclude up to 9 service classes from analysis by coding the name of the service classes to exclude in the **%LET EXCLUDEn** variables. For example, to exclude BATCHLOW Service Class, specify:

```
%LET EXCLUDE1 = BATCHLOW;        * EXCLUDE BATCHLOW SERVICE CLASS;
```

It is not necessary that the EXCLUDEn variables specified in numerical order, nor is it necessary that all numbers be specified. For example, you can safely specify:

```
%LET EXCLUDE6 = BATCHLOW;        * EXCLUDE BATCH LOW PRIORITY;
%LET EXCLUDE3 = ST_TASKS;        * EXCLUDE STARTED TASKS;
```

This feature is useful if you wish to temporarily "comment out" certain exclusions.

Note that you cannot exclude specific service class periods; CPExpert excludes all service classes specified, without regard to the period.

Chapter 1.2: Specifying data presentation guidance

The data presentation guidance provides rudimentary control over the reports which CPExpert produces as it analyzes Workload Manager constraints.

Chapter 1.2.1: VERBOSE variable.

The VERBOSE variable provides a control on the amount of narrative that the WLM Component lists with each rule result, or whether narratives are produced at all. Some installations prefer to produce only a summary report of findings each day, and evaluate

the detailed results when the findings are significant. Other installations wish to produce expanded findings, and evaluate the results on a daily basis.

Each "rule record" created by the WLM Component has a "level" associated with it. "Level 1" records are related to “basic” or important findings. "Level 2" and “Level 3" records are related to detailed more detailed comments or less important findings.

You can use the VERBOSE variable to control the amount of narrative, depending upon your preferences. The options with the VERBOSE variable are:

S = When **%LET VERBOSE=S;** is specified, CPExpert will produce only a summary introductory and statistics data, and a “graph” shown when the rules were produced.

C = When **%LET VERBOSE=C;** is specified,CPExpert will produce only “Level 1" information related to basic or important findings.

V = When **%LET VERBOSE=V;** is specified,CPExpert will produce verbose comments related to each WLM Component rule that was produced during the measurement interval. The verbose comments describe the rule, provide key information associated with the rule, and may provide a specific reference related to the rule.

Chapter 1.2.2: LISTRPT variable

The LISTRPT variable controls whether report classes are to be included in the service policy reports produced by CPExpert. CPExpert always lists the service policy in effect during the analysis (and any changes to the service policy).

Some organizations define a very large number of report classes, mostly for accounting purposes or capacity planning. If these report classes are included in the service policy reports produced by CPExpert, the reports can become quite lengthy. In such situations, it is desirable to include only *service classes* in the various reports produced by CPExpert. (Note that the Workload Manager does not manage report classes, but manages only service classes.)

Specify **%LET LISTRPT=Y;** if you wish CPExpert to include all report and service classes in the service policy report. Specify **%LET LISTRPT=N;** if you wish CPExpert to include only service classes in the service policy reports.

Chapter 1.2.3: POLORDER variable

The POLORDER variable controls whether CPExpert produces the service policy report in alphabetic sequence by service class period, or by goal importance. The default is to produce the service policy report in alphabetic sequence. Specify **%LET**

POLORDER=IMP; if you wish CPExpert to produce the service policy report sequenced by goal importance.

Chapter 1.2.4: WLMAST variable

The WLMAST variable controls whether CPExpert produces a block of asterisks when reporting a service class period that missed its performance goal. The block of asterisks allows users to appreciate better that the information relates to a new service class period. The default is to produce the service block of asterisks separating information regarding service class periods that missed their goal. Specify **%LET WLMAST=N;** if you wish CPExpert to suppress the block of asterisks. In either case, CPExpert will skip to a new page to separate service class periods that missed their performance goal.

Chapter 1.3: SAS Output Delivery System

Output from CPExpert is created using Basic SAS statements. This Basic SAS output is designed for a standard SAS printer (line) format. With SAS Release 8, SAS users can use the SAS Output Delivery System to create output that is formatted in Hypertext Markup Language (HTML). This output can be browsed with Internet Explorer, Netscape, or any other browser that fully supports the HTML 3.2 tag set.

The CPExpert WLM Component, DB2 Component, CICS Component, and DASD Component support the SAS ODS features.

Please reference the *CPExpert Installation Guide* for more detailed information about using the SAS ODS feature of CPExpert.

©Copyright 1996, Computer Management Sciences, Inc.

Chapter 2: Analysis Guidance Variables

The analysis guidance variables allow you to provide guidance to the WLM Component as CPExpert applies the WLM analysis rules. The CPEXPERT.USOURCE(WLMGUIDE) module contains defaults for each guidance variable. These defaults may be appropriate for the analysis performed by the WLM Component. However, you may have unique situations (or you may simply disagree with the defaults selected).

This chapter describes the analysis guidance variables and their defaults. **Do not hesitate to make changes if the defaults for the analysis guidance variables do not meet your needs. Please contact Computer Management Sciences if the guidance variables are inadequate for your needs.**

Exhibit 2-2 illustrates the portion of CPEXPERT.USOURCE(WLMGUIDE) that contains the analysis guidance variables.

```
**** ANALYSIS GUIDANCE VARIABLES                                          ;
%LET AVGRESP   = ELAPSED    ; * COMPUTE RESPONSE TIME USING ELAPSED TIME  ;
%LET CHKPLCY   = Y          ; * CHECK SERVICE POLICY                      ;
%LET EXECSAMP  = 100        ; * MINIMUM EXECUTION SAMPLES FOR ANALYSIS    ;
%LET HIGHCPU   = 75%        ; * PERCENT TO USE IN REPORTING HIGH CPU USERS ;
%LET INACTIVE  = 0          ; * SWITCH: INACTIVE SERVICE CLASS PERIODS     ;
%LET MAXDUR    =            ; * MAXIMUM DUR VALUE FOR TSO PERIOD 1         ;
%LET MAXRESP   = 0:05:00    ; * MAXIMUM RESPONSE GOAL                      ;
%LET MAXVEL    = 20         ; * MAXIMUM EXECUTION VELOCITY FOR BATCH       ;
%LET MINSAMP   = 100        ; * IGNORE SRV CLASS (LESS THAN 100 SAMPLES)   ;
%LET MINTRANS  = 10         ; * IGNORE INTERVALS (LESS THAN 10 TRANS.)     ;
%LET OKPAGEIN  = 5          ; * ACCEPTABLE PAGE-IN DELAY                  ;
%LET PCTSERVC = 50          ; * CONTROLS CPU PROTECTION ANALYSIS          ;
%LET PCTSERVS = 50          ; * CONTROLS STORAGE PROTECTION ANALYSIS      ;
%LET PERFINDX  = 1.0        ; * PERF INDEX-MISSED GOALS EXCEED THIS       ;
%LET PHASE     = EXECUTION  ; * ANALYZE WORK MANAGER EXECUTION PHASE       ;
%LET POLCHG    = 3          ; * MAXIMUM SERVICE POLICY CHANGES PER DAY     ;
%LET SWAPDW    = 1.0        ; * UNACCEPTABLE SWAP RATE, DETECTED WAIT      ;
%LET SWAPTO    = 0.1        ; * UNACCEPTABLE SWAP RATE, TERM OUTPUT WAIT   ;
%LET WLMnnn    = OFF        ; * EXAMPLE: TURN OFF WLMnnn RULE              ;
%LET WLMSIG    = 10         ; * SIGNIFICANT PERCENT DEGRADATION            ;
```

SAMPLE DISPLAY OF prefix.CPEXPERT.USOURCE(WLMGUIDE) MODULE

EXHIBIT 2-2

Chapter 2.1: AVGRESP variable

The AVGRESP variable is optional. The AVGRESP variable provides guidance to the WLM Component about which measure of response to use when evaluating acceptable *average* transaction response: (1) the "standard" measure of average response based upon average transaction **elapsed** time or (2) a measure of average response based upon average transaction **active** time.

The AVGRESP variable applies only when CPExpert is analyzing service classes which have an *average* response time goal. For percentile response performance goals, SMF Type 72 records provide the number of transactions completing within the performance goal, at different percentages. CPExpert can directly evaluate whether service classes meet the percentile goal based on the available information.

The default measure of transaction response time is based upon average transaction elapsed time (R723CTET), divided by the count of ending transactions (R723CRCP). This is the measure of average response time that is used by the Workload Manager and is the measure produced in RMF's *Workload Activity Report*.

The transaction elapsed time includes time swapped out in Long Wait state and may include conversation delays, depending upon the nature of the transaction. This time also includes, for example, time spent enqueued for a resource. These times can be extremely long in certain circumstances. A few transactions experiencing such long delays can completely skew the average transaction response calculations.

Some users of CPExpert wish to exclude such long wait times from the calculation of transaction response time. CPExpert permits this by optionally using transaction **active** time (R723CTAT) divided by the count of ending transactions (R723CRCP) as the measure of transaction response time.

You need make no specification if you wish to use the default (average response is transaction elapsed time divided by count of ending transactions). If you wish CPExpert to use the transaction active time divided by the count of ending transactions, specify:

```
%LET AVGRESP = ACTIVE ; * USE TRANSACTION ACTIVE TIME ;
```

Chapter 2.2: CHKPLCY variable

The CHKPLCY variable is optional. The CHKPLCY variable allows you to tell CPExpert to discontinue checking the service policy for potential problems.

The Service Policy Findings are rules in the WLM001 to WLM050 range. These findings help identify problems or potential problems with the Workload Manager service definition.

It is important to realize that these findings normally identify a POTENTIAL problem. Your systems programming staff must decide whether the findings (and their associated recommendations) make sense in your environment. For example, your systems programming staff might have deliberately selected certain parameter values. The values might be appropriate for your installation and your management objectives, even though CPExpert might produce a rule indicating that there is a potential problem with the parameter.

After you have reviewed the results of CPExpert's analysis, you may feel that the findings do not apply to your organization. It would be annoying to have inappropriate or spurious findings produced each time the WLM Component was executed. You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is set to N, CPExpert will not check the service definition for potential problems.

Before you globally disable CPExpert's checking the service definition, you may wish to review other guidance variables. Many of the tests which CPExpert makes can be made inoperative by a guidance variable that applies to the specific test. The discussion of each finding describes the associated guidance variable.

Chapter 2.3: EXECSAMP variable

The EXECSAMP variable is optional. The EXECSAMP variable is used to specify the minimum number of acceptable execution samples. The analysis performed by the Workload Manager and subsequent analysis by CPExpert is based on samples. The reliability of sampling depends upon having a sufficiently large number of samples such that the samples represent the "population" being sampled. If a small number of samples are taken, invalid conclusions might be reached based on an analysis of the samples.

If CPExpert determines that an unacceptably small number of samples exist, Rule WLM170, Rule WLM171, or Rule WLM172 is produced to indicate that no further analysis is being done. The default number of samples below which CPExpert ceases analysis is 100 samples per minute for a service class period. You can use the EXECSAMP variable to alter this default. For example, if you wish to change the unacceptably small number of samples to 50 samples per minute, specify:

```
%LET EXSAMP = 50 ; * MINIMUM NUMBER OF EXECUTION SAMPLE;
```

Chapter 2.4: HIGHCPU variable

The HIGHCPU variable is optional. CPExpert compares the amount of time when the average transaction was using the CPU against the response goal for the service class. CPExpert produces Rule WLM200 if the CPU use per transaction is **higher** than the response goal. Otherwise, CPExpert produces Rule WLM201 if the CPU use per transaction is more than 75% of the response goal.

The 75% was chosen arbitrarily as the default value, with the belief that you should be aware of such a significant amount of CPU use per transaction. You may find that the transactions naturally use a significant amount of CPU (rather than performing I/O or experiencing other delays).

You can alter the 75% default by using the HIGHCPU guidance variable. For example, to change the default value for the HIGHCPU guidance variable to 90%, specify:

```
%LET HIGHCPU = 90% ; * SERVICE CLASS HIGH CPU USE VALUE ;
```

Chapter 2.5: INACTIVE variable

The INACTIVE variable is optional. The INACTIVE variable is used to specify whether CPExpert should report situations in which service class periods were inactive for extended intervals.

CPExpert produces Rule WLM025 when any service class period was inactive for more than 75% of the RMF intervals being analyzed, and when this condition was true for all systems in the sysplex being analyzed. CPExpert ignores service class periods with discretionary goals, and ignores system service classes. Please refer to Rule WLM025 for a discussion of the rationale for this finding.

Some CPExpert users may not care that the service class period is inactive, and do not wish to be annoyed by Rule WLM025 being produced spuriously. You can use the INACTIVE variable to suppress this analysis and finding, by specifying:

```
%LET INACTIVE = N; * SUPPRESS ANALYSIS OF INACTIVE SERVICE CLASSES;
```

©Copyright 1996, Computer Management Sciences, Inc.

Chapter 2.6: MAXDUR variable

The MAXDUR variable is optional. The MAXDUR variable is used to specify the unacceptable period duration for TSO Period 1 service classes. The MAXDUR variable is used to guide RULE WLM008.

The normal purpose of defining multiple service class periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources.

The SRM will be able to differentiate between interactive and non-interactive transactions only if the values specified for the DUR keyword roughly correspond to the resource requirements of trivial, interactive, and non-interactive transactions.

If the value specified for the DUR keyword for Period 1 is too large, non-trivial and non-interactive transactions will execute with the same performance management controls as those given to trivial transactions. This would defeat the purpose of breaking the service class into multiple service class periods.

CPExpert concludes that the DUR value is too large for TSO Period 1 transactions if the DUR value is greater than 100 (for service policies which have MSO less than 1) or greater than 300 (for service policies which have MSO equal to 1 or higher). This conclusion is based upon comparing the DUR value with the DUR values specified by other installations.

You may have unique requirements for TSO Period 1 transactions executing at your installation, or you may not agree with CPExpert's conclusions. You can use the MAXDUR guidance variable to change the DUR value which CPExpert considers to be high. For example, if you wish to change the guidance to indicate that the Rule WLM008 should be produced when the DUR value is greater than 1000, specify:

```
%LET MAXDUR = 1000 ; * ACCEPTABLE TSO PERIOD 1 DUR VALUE;
```

Chapter 2.7: MAXRESP variable

The MAXRESP variable is optional. The MAXRESP variable is used to specify the unacceptable response performance goal. The MAXRESP variable is used to guide RULE WLM006.

The Workload Manager ISPF Response Time Goal Panel allows a response performance goal of up to 24 hours to be specified. Response goals in minutes or hours are typically associated with batch workloads.

CPExpert believes that a response performance goal of over 5 minutes is likely to result in unsatisfactory performance in most environments and a response goal of **less than 1 minute** is more likely to yield desired results. Rule WLM006 describes the reasoning behind this belief.

CPExpert produces Rule WLM006 if a response performance goal is greater than the **MAXRESP** guidance variable in USOURCE(WLMGUIDE). The default value for the MAXRESP guidance variable is 0:05:00, indicating that Rule WLM006 will be produced if a response goal of more than 5 minutes is defined.

If you believe that your specification is appropriate, you can use the MAXRESP guidance variable to provide guidance to CPExpert. For example, if you wish to change the guidance to indicate that the Rule WLM006 should be produced when the response is greater than 30 minutes[2], specify:

```
%LET MAXRESP = 0:30:00 ; * ACCEPTABLE RESPONSE PERFORMANCE GOAL;
```

Chapter 2.8: MAXVEL variable

The MAXVEL variable is optional. The MAXVEL variable is used to specify the unacceptable execution velocity performance goal **for service classes describing batch jobs**. The MAXVEL variable is used to guide RULE WLM005.

A high execution velocity goal can cause significant system problems if there is the possibility of batch work (or other types of work, for that matter) to be erratic in nature. For example, if batch jobs can enter into a CPU loop, the batch workload may "seize" the system for whatever percentage was specified as the execution velocity. To illustrate, suppose that an execution velocity of 50 was specified for a batch service class. A CPU-intensive batch job (or CPU-looping job) could require 50% of the CPU and deny CPU access to all work of lower dispatching priority.

[2]**Please note that CPExpert strongly feels that such a specification is incorrect!** The MAXRESP guidance variable is provided only for those installations who insist on specifying a large response goal and who would be annoyed by CPExpert constantly providing advice that the response goal is too large.

IBM Workload Manager developers have suggested that an execution velocity of 10 or 20 should be adequate for most batch service classes. These values are based on their observations that most batch jobs are naturally I/O intensive, rather than CPU intensive.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "batch" and assumes that the service class describes batch workload if "batch" is encountered.

CPExpert produces Rule WLM005 if an execution velocity performance goal is greater than the **MAXVEL** guidance variable in USOURCE(WLMGUIDE). The default value for the MAXVEL guidance variable is 20, indicating that Rule WLM005 will be produced if more than 20 had been specified as an execution velocity goal for a service class containing batch workload.

If you believe that your specification is appropriate, you can use the MAXVEL guidance variable to provide guidance to CPExpert. For example, if you wish to change the guidance to indicate that the Rule WLM005 should be produced when the execution velocity for batch jobs is greater than 30, specify:

```
%LET MAXVEL = 30 ; * ACCEPTABLE BATCH EXECUTION VELOCITY;
```

Since 99 is the maximum execution velocity specification for the Workload Manager, you can disable Rule WLM005 by specifying a value of 99 for the MAXVEL guidance variable (that is, specify %LET MAXVEL=99;).

Chapter 2.9: MINSAMP variable

The MINSAMP variable is optional. The MINSAMP variable specifies the minimum number of work "using or delay" samples for a service class in any RMF measurement interval. The problems caused by only a few samples in a recording interval are described in Chapter 5 of Section 4[3].

The default value for the MINSAMP variable is 100. You can change this value if you wish CPExpert to use a different number of samples. For example, to ignore a service class data in any RMF recording intervals containing less than 200 samples, specify:

[3]Briefly, the "using and delay" values are sampled values taken by the Workload Manager approximately every 250 milliseconds. Only a few "using or delay" samples may be taken if a service class has an address space active for a short time in any RMF measurement interval. If only a few samples are taken, conclusions based on the samples may be invalid.

```
%LET MINSAMP = 200 ; * MINIMUM USING OR DELAY SAMPLES;
```

RMF recording intervals with transactions ending below this minimum are considered to represent an invalid RMF measurement interval from a workload and performance analysis view. The WLM Component will ignore the service class for these RMF recording intervals, with the assumption that the samples are too few to warrant conclusions. The WLM Component will report any recording intervals rejected for this reason. We recommend that you set the MINSAMP variable to at least 100.

Chapter 2.10: MINTRANS variable

The MINTRANS variable is optional. The MINTRANS variable specifies the minimum number of transactions for a service class in any RMF measurement interval. The problems caused by only a few transactions in a recording interval are described in Chapter 5 of Section 4[4].

The default value for the MINTRANS variable is 10. You can change this value if you wish CPExpert to use a different number of samples. For example, to ignore a service class data in any RMF recording intervals containing less than 50 transactions, specify:

```
%LET MINTRANS= 50 ; * MINIMUM ENDING TRANSACTIONS;
```

Chapter 2.11: OKPAGEIN variable

The OKPAGEIN variable is optional. The OKPAGEIN variable is used to specify the value CPExpert should use to assess whether page-in imbalance is significant; page-in delay exceeding the average page-in delay may cause CPExpert to produce Rule WLM058 (local page data set response is significantly worse than average).

When a service class period misses its performance goal and page-in delay is a major cause of delay, CPExpert analyzes the paging subsystem. One potential cause of page-in problems is an imbalance in the paging operations from a particular local page data set.

[4]Briefly, many of the algorithms which the Workload Manager uses depend upon a reasonable number of ending transactions. Likewise, CPExpert's analysis depends upon a reasonable number of transactions. If too few transactions end in any RMF measurement interval, conclusions based on the RMF interval may be invalid.

Typically, these problems are caused by controller or device contention of some sort. Please refer to Rule WLM058 for a discussion of the rationale for this finding.

CPExpert computes the average page-in time for all local page data sets. CPExpert produces Rule WLM058 to identify any local page data set having a page-in time more than 50% greater than the average page-in delay for all local page packs on the system.

Some users might wish to have a minimum average page-in delay value, before CPExpert applies the algorithm. The **OKPAGEIN** guidance variable in USOURCE(WLMGUIDE) can be used to provide guidance to CPExpert regarding the minimum average page-in delay you consider acceptable. For example, if you wish to change the guidance to indicate that Rule WLM058 should be produced unless the average page-in delay is greater than 5 milliseconds, specify:

```
%LET OKPAGEIN = 5; * ACCEPTABLE AVERAGE PAGE-IN DELAY;
```

Chapter 2.12: PCTSERVC variable

The PCTSERVC variable is optional. The PCTSERVC variable is used to specify the value CPExpert should use to assess whether most CPU work in the CICS or IMS region was done in support of lower importance transaction service classes, when a server (CICS or IMS region) was assigned long-Term CPU Protection.
It is common for a server to process transactions that have been assigned to more than one transaction service class, since some transactions are very important while other transactions are less important. The transactions can be classified and assigned to different transaction service classes, and the transaction service classes can have different performance goals and goal importance. If transactions in these different transaction service classes are processed by a server with CPU protection assigned, all transactions receive the same CPU protection, regardless of their importance.

CPExpert produces Rule WLM032 when the total service provided to the transaction service classes at the highest goal importance is less than the value specified for the **PCTSERVC** guidance variable in USOURCE(WLMGUIDE). The default value for the PCTSERVC guidance variable is 50%, indicating that Rule WLM032 would be produced when the most important transaction service classes received less than 50% of the service provided by the server to all transaction service classes. You can alter CPExpert's analysis by modifying the PCTSERVC guidance variable in USOURCE(WLMGUIDE).
For example, if you wish to change the guidance to indicate that Rule WLM032 should not be produced unless more than 75% of the service provided by a server (IMS or CICS region) was provided to low importance work , specify:

```
%LET PCTSERVC = 75; * ACCEPTABLE USE OF SERVER BY LOW IMPORTANCE;
```

Chapter 2.13: PERFINDX variable

The PERFINDX variable is optional. CPEXPERT computes an average Performance Index for each service class with a performance goal[5], for each RMF measurement interval selected for analysis. CPExpert compares the computed average Performance Index against the PERFINDX guidance variable. If the computed average Performance Index is greater than the PERFINDX guidance variable, CPExpert analyzes the service class and attempts to determine why the performance goal was not met.

In many environments, performance of a service class will be very close to the performance goal. In fact, the Workload Manager does not consider performance to be worthy of action unless the Performance Index is 1.1 (indicating that about 10% of the performance goal was not achieved). After executing the WLM Component of CPExpert for some time, you may find that you wish CPExpert to perform detailed analysis only when performance was **significantly** worse than the performance goal.

The default value for the PERFINDX guidance variable is 1.0, indicating that CPExpert will perform a detailed analysis of any service class that failed to achieve its performance goal for an **entire** RMF measurement interval.

You can alter the performance at which CPExpert begins its analysis of a service class by using the PERFINDX guidance variable. For example, to change the default value for the PERFINDX guidance variable to 1.1%, specify:

```
%LET PERFINDX = 1.1; * PERFORMANCE INDEX-MISSED GOALS EXCEED THIS;
```

[5]Please refer to Section 4 for a discussion of the Performance Index used by the Workload Manager. Briefly, the performance index is a measure of how well the service class met its performance goal. If the Performance Index is less than 1, the service class had performance better than its goal. If the Performance Index is exactly 1, the service class exactly met its performance goal. If the Performance Index is greater than 1, the service class had performance worse than its goal. The performance of a number of service classes can be compared by simply comparing the relative value of their Performance Indexes.

Chapter 2.14: PHASE variable

The PHASE variable is optional. The PHASE variable provides guidance to CPExpert about how to analyze "served" service classes. SMF provides information about delays in transaction *begin_to_end phase* and information about delays in the transaction *execution* phase[6].

- The *begin_to_end phase* includes the total time a transaction is active from the perspective of the system (it does not include network delays at the start and end of the transaction). This time can include waiting for interactive conversations, waiting for functions shipped to another system, etc.

- The *execution phase* includes the time in which transactions were active in the system being analyzed (that is, the transactions were not waiting on external events such as conversations).

Many users of CPExpert will wish to analyze transaction delays during the execution phase. This is because the begin-to-end phase includes such waits as "waiting for conversations" which are outside the control of the performance analyst. If the begin-to-end phase data were used to analyze delays to response time, the results might be skewed toward areas which cannot be easily reduced. Thus, the default analysis phase is the execution phase.

However, you can use the **PHASE** guidance variable to cause CPExpert to analyze transaction delays from the perspective of begin-to-end phase data. If you wish CPExpert to use the begin-to-end phase data in analyzing transaction delays[7], specify:

```
%LET PHASE = BEGIN_TO_END ; * USE BEGIN_TO_END PHASE DATA ;
```

Chapter 2.15: POLCHG variable

The POLCHG variable is optional. The POLCHG variable is used to specify the number of acceptable service policy changes per day. The POLCHG variable is used to guide RULE WLM004.

---

[6]Please refer to Section 4 for a more complete discussion of the begin_to_end phase and execution phase.

[7]Please note that the begin-to-end phase analysis is very modest in the initial implementation of the WLM Component of CPExpert; basically, the analysis is restricted to the information available in SMF Type 72 records. This is because there is little information currently available about function shipping within a sysplex and the problems encountered.

As CPExpert users begin to implement more sysplex-based capabilities, CPExpert will be enhanced to provide better analysis. Please give Computer Management Sciences a call at (703) 922-7027 if you begin to use sysplex function shipping and find that the begin-to-end analysis done by CPExpert is inadequate. We will be delighted to work with you to improve our product!

Service policy changes should occur relatively infrequently, as any service policy change causes some adverse effect on performance simply because the policy changed. Please refer to Rule WLM003 for a discussion of service policy changes and the harmful effect on system performance of the changes. A relatively large number of changes normally should be cause for management action to reduce the changes.

CPExpert accumulates the number of service policy changes. CPExpert produces Rule WLM004 if the number of changes per day is greater than the **POLCHG** guidance variable in USOURCE(WLMGUIDE).

The default value for the POLCHG guidance variable is 3, indicating that no more than 3 policy changes should occur during any interval being analyzed by CPExpert, or 3 changes per day. If the data being analyzed by CPExpert covers more than one day, CPExpert normalizes the number of changes to a per-day basis.

You may have a unique situation in which you must make more than 3 policy changes (or you may wish to be notified of **any** policy changes). The **POLCHG** guidance variable in USOURCE(WLMGUIDE) can be used to provide guidance to CPExpert regarding the number of service policy changes you consider acceptable. For example, if you wish to change the guidance to indicate that Rule WLM004 should be produced when more than 5 policy changes occur per day, specify:

```
%LET POLCHG = 5 ; * ACCEPTABLE NUMBER OF POLICY CHANGES PER DAY;
```

Chapter 2.16: SWAPDW variable

The SWAPDW variable is optional. The SWAPDW variable is used to specify the unacceptable swap rate for Detected Wait swaps. The SWAPDW variable is used to guide RULE WLM071.

Detected Wait swaps occur because the SRM detects that a resident transaction has not been dispatchable for two seconds of real time or eight SRM seconds, without issuing the WAIT,LONG=YES macro. Detected Wait swaps usually are caused by cross memory services, applications that treats the terminal as SYSIN or SYSPRINT, teleprocessing applications (e.g., test CICS regions) that are not marked non-swappable, etc.

Detected Wait swaps depend upon the applications executing in your environment. For example, if you have a large number of test CICS regions marked as swappable, these regions will be swapped out with a Detected Wait swap.

©Copyright 1996, Computer Management Sciences, Inc.

You **normally should alter the default value** for the SWAPDW guidance variable. You should change the guidance to a different value, depending upon the application mix executing at your installation. For example, if you wish to change the guidance to indicate that the swap rate for Detected Wait swaps is unacceptable when these swaps occur more often than once per second, specify:

```
%LET SWAPDW = 1 ; * UNACCEPTABLE SWAP RATE: DETECTED WAIT;
```

Chapter 2.17: SWAPTO variable

The SWAPTO variable is optional. The SWAPTO variable is used to specify the unacceptable swap rate for Terminal Output Wait swaps. The SWAPTO variable is used to guide RULE WLM070.

Terminal Output Wait swaps occur after the SRM has been notified that a TSO session is in terminal wait after issuing a TPUT, and the address space is in a long wait condition. Please refer to Rule WLM070 for a discussion of Terminal Output Wait swaps and ways to prevent these swaps.

RULE WLM070 will be produced if the below conditions apply:

- Any service class missed its performance goal and a significant cause of the delay was swap-in delay or MPL delay[8].

- The Terminal Output Wait swap rate is greater than the value specified by the SWAPTO guidance variable.

Most analysts believe that there is little reason to experience more than an occasional Terminal Output Wait swap if proper values are used for HIBFREXT and LOBFREXT keywords in the TSOKEYxx member of SYS1.PARMLIB. Consequently, the default value for the SWAPTO variable is set to .01 - indicating that more than one Terminal Output Wait swap every 100 seconds is unacceptable.

You can alter the default if you wish to change the guidance to a different value. For example, if you wish to change the guidance to indicate that the swap rate for Terminal Output Wait swaps is unacceptable when these swaps occur more often than once per 50 seconds, specify:

[8]Please refer to Section 4 for a discussion of these delays.

%LET SWAPTO = .02 ; * UNACCEPTABLE SWAP RATE: TERM OUTPUT;

Chapter 2.18: Turning OFF WLM Component Rules

The default guidance values for the WLM Component are specified based on either IBM's guidance contained in IBM documents, or based on guidance from industry sources. For most findings, users can use analysis guidance variables to control the analysis; rule results will be produced only when situations exceed the guidance provided.

However, some users of the WLM Component wish to suppress the analysis and findings of particular rules. This desire typically is caused by (1) an overall disagreement with the finding, (2) an inability to make a suggested change, or (3) a decision that a particular finding is inapplicable to a particular system.

For example, Rule WLM012 (A server defaulted to the SYSSTC service class) might not be applicable to particular installations or you might disagree with the reasoning.

Regardless of the reason for wishing to suppress particular findings by the WLM Component, users wish the ability to "turn off" certain rules.

All rules are ON by default, although the WLM Component may turn rules OFF if insufficient data exists to perform analysis or if the rule does not apply to the version of OS/390 being analyzed.

Rules can be turned OFF by specifying **%LET WLMnnn = OFF;** , where "nnn" is the rule number that you wish to turn OFF. This specification should be placed in the *ANALYSIS GUIDANCE SECTION* of USOURCE(WLMGUIDE).

For example, you can turn OFF Rule WLM012 by specifying **%LET WLM012=OFF;** in the *ANALYSIS GUIDANCE SECTION* of USOURCE(WLMGUIDE).

If you are analyzing coupling facility performance constraints during a single execution of the WLM Component and you are specifying specific guidance for particular coupling facility structures, you can **not** turn rules OFF (or turn rules ON) for particular structures.

Please note that the WLM Component normally verifies that all required data is present in your performance data base before invoking each rule. A rule will be suppressed if any required data is missing, regardless of your specification to suppress or enable the rule.

©Copyright 1996, Computer Management Sciences, Inc. Revised: April, 2003

Chapter 2.19: WLMSIG variable

The WLMSIG variable is optional. The WLMSIG variable specifies the percent of delay which causes a rule to be produced. Each service class might experience some time in most, if not all, possible causes of delay. From one perspective, **any** time spent in a delay category is "degradation" caused by that delay category.

However, if the WLM Component produced a rule regardless of the percent of delay related to some delay category, many WLM Component rules might be produced. The result would not be particularly meaningful; the reports would be cluttered by findings which have little impact on performance and which would report on areas in which it would not be worth expending effort to reduce the delay. Rather, the findings should reflect delays only when a particular delay was a **significant** factor in a service class failing to achieve its performance goal.

The WLMSIG variable allows control of the definition of "significant" delay. The WLMSIG value is specified as a percent against which various delay computations are compared. If the delay percentage is **less** than WLMSIG, the corresponding rule is suppressed. If the delay is **equal to or greater than** WLMSIG, the corresponding rule is produced.

For example, to specify that rules should be produced only if a particular delay accounted for more than 30% of unacceptable response time, specify:

```
%LET WLMSIG = 30 ;          * SIGNIFICANT PERCENT DELAY;
```

Chapter 3: XCF Performance Analysis Guidance Variables

The cross system coupling facility (XCF) component of MVS/ESA SP5 (and above) allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or with programs on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

The coupling facility is licensed internal code running in a special type of PR/SM logical partition (LPAR). The coupling facility is used for data sharing across systems in a sysplex, maintains the integrity and consistency of the shared data, and maintains the availability of a sysplex. A coupling facility includes the licensed internal code, and includes processor storage (central storage and perhaps expanded storage).

Storage in the coupling facility primarily is divided into *XCF structures* that are further identified as three types: cache, list, and lock. Authorized programs use in the coupling facility to implement data sharing and serialization. Additionally, some storage in the coupling facility normally is allocated as a dedicated dump space for capturing structure information for diagnostic purposes.

Within the XCF terminology, authorized programs are termed *XCF members*, and the XCF members are logically a part of specific *XCF Groups*. For example, CICS regions are considered XCF members, and the regions are logically associated with the **DFHIR000** XCF Group. RMF is logically associated with the **SYSRMF** XCF Group, the MVS Workload Manager is associated with the **SYSWLM** XCF Group, etc. One purpose of associating members with XCF groups is to facilitate system management control for similar applications.

XCF group members communicate with each other using the XCF *signaling* mechanism. The communication is done via signaling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signaling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

The above two areas (structures and signaling) are the main areas to evaluate when considering the performance of a sysplex. Initially, the XCF performance analysis done by CPExpert is limited to analyzing the performance of the signaling service. This limit is because the performance implication of signaling options is better understood than is the performance implication of structure options.

Different XCF groups have different signaling characteristics and different signaling performance requirements.

©Copyright 1996, Computer Management Sciences, Inc.

- For example, the Workload Manager group (SYSWLM) sends a message approximately every 10 seconds. The message is 300 bytes * the number of service class periods with a response time or velocity goal. For a typical installation, this message might be less than 5,000 bytes. Although it is *desirable* that the Workload Manager have up-do-date information, it is not *critical* that the SYSWLM message be received at once.

- On the other hand, global resource serialization (GRS) sends such messages as the RSA-message to provide information about the serialization of global resources. The RSA-message can be sent frequently, and can be up to 32K bytes of data. It is critical to the performance of applications that the GRS message be received at once.

Optimal signaling performance requires that XCF groups have access to adequate signaling resources. These resources consist of signaling paths and buffers. Since different XCF groups have different signaling requirements, performance usually is improved if signaling resources are assigned to the XCF groups based on their requirements.

A *transport class* is the mechanism used by MVS to allow resources to be assigned to XCF groups. Resources (signaling paths, buffers, etc.) are assigned to one or more transport classes, and XCF groups are assigned to the transport classes. Thus, resources can be made available to the XCF groups as they are needed.

A particular MVS system has limited resources, and not all XCF groups require the same amount of resources. Consequently, one performance tuning consideration is the balance between (1) the resources available, (2) the resources required by different XCF groups, and (3) the value (or importance) to the installation of the various XCF group members.

CPExpert's XCF analysis guidance variables allows you to guide CPExpert's analysis of signaling performance. Default thresholds have been established based on information contained in IBM publications and other documents. These defaults may not be suitable for your environment and specific management objectives. If the analysis and reports produced by CPExpert do not meet your needs, alter the guidance to CPExpert. If the guidance is insufficient, please call Computer Management Sciences at (703) 922-7027 so we can make changes to improve CPExpert for you!

Exhibit 2-3 illustrates the USOURCE(WLMGUIDE) variables that provide guidance to the WLM Component as it analyzes XCF performance. This chapter describes these variables, how the variables are used, and how the variables are altered.

©Copyright 1996, Computer Management Sciences, Inc.

```
*********************************************************************      ;
*          WLM COMPONENT GUIDANCE VARIABLES                    ;
*********************************************************************      ;
          •
          •
          •
          •
**** CROSS SYSTEM COUPLING FACILITY GUIDANCE VARIABLES   ;
%LET COUPLE          = N ;     * SWITCH: CONTROLS COUPLING FACILITY ANALYSIS;
%LET EXCLASS1        =;        * EXCLUDE TRANSPORT CLASS 1          ;
%LET EXCLASS2        =;        * EXCLUDE TRANSPORT CLASS 2          ;
%LET EXCLASS3        =;        * EXCLUDE TRANSPORT CLASS 3          ;
%LET EXCLASS4        =;        * EXCLUDE TRANSPORT CLASS 4          ;
%LET EXCLASS5        =;        * EXCLUDE TRANSPORT CLASS 5          ;
%LET EXCLASS6        =;        * EXCLUDE TRANSPORT CLASS 6          ;
%LET EXCLASS7        =;        * EXCLUDE TRANSPORT CLASS 7          ;
%LET EXCLASS8        =;        * EXCLUDE TRANSPORT CLASS 8          ;
%LET EXCLASS9        =;        * EXCLUDE TRANSPORT CLASS 9          ;
%LET PCTSML          = 90;     * ACCEPTABLE % SMALL MESSAGES        ;
%LET PCTBIG          = 1;      * ACCEPTABLE % BIG MESSAGES          ;
%LET PCTREJ          = .1;     * ACCEPTABLE % MESSAGES REJECTED, NO BUFFER SPACE;
%LET LOCKCONT        = 2;      * MAXIMUM PERCENT LOCK CONTENTION;
%LET FALSECNT        = .5;     * MAXIMUM PERCENT FALSE LOCK CONTENTION;
%LET SYNCSRV         = 350;    * ACCEPTABLE SERVICE TIME (MICROSEC) SYNCH REQ;
%LET LOCKSRV         = 250;    * ACCEPTABLE SERVICE TIME (MICROSEC) LOCK REQ;
%LET ASYNCSRV        = 5000;   * ACCEPTABLE SERVICE TIME (MICROSEC) ASYNCH REQ;
%LET SYNCCHG         = 10;     * ACCEPTABLE PCT CHANGED (SYNCH TO ASYNCH);

%LET STRGUIDE        = Y;      * GUIDANCE IS PROVIDED FOR INDIVIDUAL STRUCTURES;
/* SPECIFY GUIDANCE FOR STRUCTURES
STRUCTURE = structure.name1
LOCKCONT =;          * MAXIMUM PERCENT LOCK CONTENTION;
SYNCSRV =  ;         * ACCEPTABLE SERVICE TIME, SYNCHRONOUS REQ;
LOCKSRV =  ;         * ACCEPTABLE SERVICE TIME, LOCK REQUESTS;
ASYNCSRV =;          * ACCEPTABLE SERVICE TIME, ASYNCHRONOUS REQ;
SYNCCHG  = ;         * ACCEPTABLE PCT CHANGED (SYNCH TO ASYNCH)  ;
*/
*********************************************************************    ;
```

DEFAULT VALUES FOR XCF ANALYSIS IN USOURCE(WLMGUIDE)

EXHIBIT 2-3

Chapter 3.1: COUPLE variable

The COUPLE guidance variable is used to tell CPExpert whether you wish to analyze cross system coupling facility performance. If the COUPLE variable is **N**, CPExpert will not analyze XCF performance. If the COUPLE variable is **Y**, CPExpert will analyze XCF performance.

The default specification for the COUPLE guidance variable is **N**, indicating that CPExpert should not analyze coupling facility data. This default was selected because few installations currently operate in an XCF environment. If you wish CPExpert to analyze XCF performance, specify:

```
%LET COUPLE = Y ; * SWITCH: CONTROLS COUPLING FACILITY ANALYSIS;
```

Chapter 3.2: EXCLASSW and EXCLASSn variables

For a variety of reasons, you may wish to exclude certain transport classes from analysis. The most common reason is that CPExpert repeatedly identifies problems with a particular transport class, but (1) you do not wish to make changes to correct the problems, (2) you are unable to make changes (because of application requirements or political realities), or (3) the WLM Component analysis is "flawed" because of data problems or data averaging. Whatever the reason, you may wish to exclude certain transport classes from analysis.

Excluding transport classes from analysis is optional; you do not have to exclude any transport classes from analysis. If you do **not** exclude transport classes from analysis, CPExpert will analyze every transport class encountered in SMF Type 74 data.

If you wish to exclude transport classes from analysis, you must (1) use the **EXCLASSW** guidance variable and (2) define the transport classes to exclude.

Chapter 3.2.1: EXCLASSW variable

The EXCLASSW guidance variable is used to tell CPExpert whether you wish to exclude transport classes from analysis. The EXCLASSW variable acts as a "switch" to control whether CPExpert excludes transport classes from analysis. The point of having a "switch" variable is that some installations may wish to regularly exclude transport classes from analysis, but periodically analyze all transport classes.

If the EXCLASSW variable is **N**, CPExpert will not exclude any transport classes from analysis. If the EXCLASSW variable is **Y**, CPExpert will exclude all transport classes specified by the EXCLASSn guidance variable(s).

©Copyright 1996, Computer Management Sciences, Inc.

Chapter 3.2.2: EXCLASSn variable(s)

You may exclude up to 9 transport classes from analysis by coding the name of the transport classes to exclude in the **%LET EXCLASSn** variables. For example, to exclude the DEFAULT Transport Class, specify:

```
%LET EXCLASS1 = DEFAULT;   * EXCLUDE DEFAULT TRANSPORT CLASS;
```

It is not necessary that the EXCLASSn variables specified in numerical order, nor is it necessary that all numbers be specified. For example, you can safely specify:

```
%LET EXCLASS6 = DEFAULT;      * EXCLUDE DEFAULT TRANSPORT CLASS;
%LET EXCLASS3 = CICSCLAS;     * EXCLUDE CICSCLAS TRANSPORT CLASS;
```

This feature is useful if you wish to temporarily "comment out" certain exclusions.

Chapter 3.3: PCTSML variable

Outbound message buffers are assigned to transport classes in two ways: (1) the basic assignment to the transport class via the CLASSLEN and MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

- The CLASSLEN parameter defines the message length for the transport class. MVS allocates *fixed-length buffers* at the size specified in the CLASSLEN parameter for the transport class.

- The MAXMSG parameter defines the amount of message buffer space allocated for messages sent in the transport class. The MAXMSG parameter can be specified on the PATHOUT or PATHIN statements, or on the CLASSDEF statement.

The message length specified by the CLASSLEN parameter should be large enough to accommodate most messages, but not so large as to waste storage. Selecting the correct buffer length is a tradeoff between (1) overhead incurred by having buffers too small, (2) wasted storage incurred by having buffers too large, and (3) the performance implications of mixing large and small messages in the same transport class.

- If the fixed-length buffers are too small to hold a message, MVS acquires additional buffers to accommodate the message. Increased system overhead is caused when MVS must acquire additional buffers. In order to minimize this overhead, MVS may

dynamically increase the length of the buffers if (1) the number of over-sized messages message traffic warrants the increase and (2) the increase in buffer length would not exceed the maximum buffer space specified on the receiving system.

- If the buffers are too large for a message, the unused storage remaining in the buffer is wasted. This is an inefficient use of storage. Additionally, MVS could exhaust the supply of buffer space associated with a transport class if the space is wasted by specifying a buffer length that is too large for most messages. In the later case, XCF messages would be rejected if the supply of buffer space is exhausted.

- If large and small messages are mixed in the same transport class, the small messages tend to be delayed simply because the large messages take longer to process.

Please refer to the discussion in Rule WLM601 to Rule WLM603 for more information about outbound message buffers.

CPExpert analyzes information in SMF Type 74 (Subtype 2) records to determine whether the correct buffer allocation has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message length specified for the transport class is too large when a *significant percent* of the messages were **smaller** than the buffer length specified for the transport class. CPExpert produces Rule WLM603 based on this conclusion.

The value considered a "significant percent" of the messages is controlled by the **PCTSML** guidance variable. If most of the outbound messages do not fit the buffer lengths, it normally is better for the buffer lengths to be slightly larger than the outbound messages. A small amount of wasted storage usually has less performance impact than the unnecessary overhead caused by messages being larger than the buffer length.

The default specification for the PCTSML variable is **%LET PCTSML = 90;**. This value for PCTSML is intended to cause Rule WLM603 to be produced when more than 90% of the messages are smaller than the defined buffer length. You can alter the analysis by specifying a different value (and you can override the analysis completely by specifying **%LET PCTSML = 100;** for the guidance). For example, if you wish to be notified when 99 percent of the outbound messages were smaller than the defined buffers, specify:

```
%LET PCTSML = 99 ; * ACCEPTABLE PERCENT SMALL MESSAGES;
```

Chapter 3.4: PCTBIG variable

As mentioned above, outbound message buffers are assigned to transport classes in two ways: (1) the basic assignment to the transport class via the CLASSLEN and MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

Please refer to the discussion in Rule WLM601 to Rule WLM603 for more information about outbound message buffers.

CPExpert analyzes information in SMF Type 74 records to determine whether the correct buffer allocation has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message length specified for the transport class is too small when a *significant percent* of the messages were **larger** than the buffer length specified for the transport class. CPExpert produces Rule WLM601 or Rule WLM602 based on this conclusion.

The value considered a "significant percent" of the messages is controlled by the **PCTBIG** guidance variable. If most of the outbound messages exceed the buffer lengths,

The default specification for the PCTBIG guidance variable is **%LET PCTBIG = 1;**, indicating that Rule WLM601 or Rule WLM602 will be produced when more than 1% of the messages were too large for the defined buffers. You can alter this analysis using the PCTBIG guidance variable. For example, if you wish to be notified when 5 percent of the outbound messages were larger than the defined buffers, specify:

```
%LET PCTBIG = 5 ; * ACCEPTABLE PERCENT BIG MESSAGES;
```

Chapter 3.5: PCTREJ variable

Outbound message buffers are used to send messages to another system. Message buffer space for **outbound** messages is separated by transport class, so a sudden high volume of traffic in one transport class will not cause performance problems for another transport class. If the message buffer space required to support messages in a particular transport class is exhausted, MVS will reject additional messages until outbound message buffer space becomes available in the transport class.

Inbound message buffers are used to receive messages from another system. These buffers are allocated, as needed, to support the message traffic load. Message buffer space for **inbound** messages is separated by signaling path. XCF obtains inbound message buffers for each inbound path before an inbound signal is necessarily sent; this

action is in *anticipation* of an inbound signal and is intended to prevent delays waiting for buffers. Message buffers associated with an inbound signaling path do not receive messages over any other inbound signaling path. If the inbound message buffer space required to support messages on a particular inbound signaling path is exhausted, MVS will reject additional messages until message buffer space becomes available in for the inbound signaling path.

Local message buffers are used to send and receive messages from programs within the same system. These buffers are allocated, as needed, to support the message traffic load. Message buffer space for **local** messages is separated by transport class, so a sudden high volume of traffic in one transport class will not cause performance problems for another transport class. If the message buffer space required to support messages in a particular transport class is exhausted, MVS will reject additional messages until local message buffer space becomes available in the transport class.

CPExpert analyzes SMF Type 74 (Subtype 2) information to determine whether sufficient message buffer space has been defined. CPExpert computes the total outbound, inbound, and local message traffic. CPExpert concludes that the message buffer space is too small when more than the value specified for the **PCTREJ** guidance variable of the messages outbound, inbound, or local messages were rejected because of no buffer space. CPExpert produces Rule WLM604, Rule WLM605, or Rule WLM606 to indicate that message buffer space may be too small for outbound, inbound, or local message buffer space, respectively.

The default specification for the PCTREJ guidance variable is **%LET PCTREJ = 0.1;** indicating that Rule WLM604-WLM606 will be produced when more than one-tenth of a percent of the message traffic is rejected for insufficient buffer space. You can alter this analysis using the PCTREJ guidance variable. For example, if you wish to be notified when 1 percent of the messages were rejected because of insufficient buffer space, specify:

```
%LET PCTREJ = 1 ; * ACCEPTABLE % MESSAGES REJECTED , NO BUFFERS;
```

Chapter 3.6: LOCKCONT variable

Locking is the mechanism used to reserve all or part of a database so that other programs will not be able to update the data until you have finished processing the data. By locking the data, users can be sure that the information they are processing is current. Without locking, users might lose updates or access invalid or incomplete data. Locking is necessary, of course, only if one or more of the users of the data will be performing updates. If no updating of the data is performed, locking is unnecessary; the data may be

concurrently accessed by any number of user without worry that the data is incomplete or invalid.

Lock contention occurs when one user wishes to access data and some other user has placed a lock on the data. The user wishing to access the data usually is suspended until the data is available (that is, until the lock is released). Techniques such as separating data, choosing locking parameters, and monitoring for contention can be used to provide a balance between concurrency of access, isolation and integrity of data, and efficient use of system resources.

The coupling facility lock structure contains information used to determine cross-system contention on a particular resource. IRLM assigns (or "hashes") locked resources to an entry value in the lock structure in the coupling facility. IRLM uses the lock table to determine whether a resource is locked. If the lock structure defined on the coupling facility is too small, the hashing algorithm can select the same lock table entry for two different locks. This situation is termed *false lock contention*. The user wishing to access the locked data is suspended until it is determined that there is no real lock contention on the resource.

CPExpert divides SMF Type 74 (Subtype 4) field R744SSCN (the number of times any request encountered lock contention) by R744STRC (the total number of lock-related requests) for lock structures, to yield the percent of requests that experienced lock contention. CPExpert compares this percentage with the **LOCKCONT** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM651 when the percent of lock contention exceeds the value specified by the LOCKCONT variable.

The default specification for the LOCKCONT guidance variable is **%LET LOCKCONT = 2;** indicating that Rule WLM651 will be produced when more than 2% of the requests experienced lock contention. You can alter this analysis using the LOCKCONT guidance variable. For example, if you wish to be notified when 1 percent of the requests experienced lock contention, specify:

```
%LET LOCKCONT = 1 ; * ACCEPTABLE PERCENT LOCK CONTENTION;
```

Chapter 3.7: FALSECNT variable

The coupling facility lock structure contains information used to determine cross-system contention on a particular resource. IRLM assigns (or "hashes") locked resources to an entry value in the lock structure in the coupling facility. IRLM uses the lock table to determine whether a resource is locked. If the lock structure defined on the coupling facility is too small, the hashing algorithm can select the same lock table entry for two different locks. This situation is termed *false lock contention*. The user wishing to access

the locked data is suspended until it is determined that there is no real lock contention on the resource.

CPExpert divides R744SFCN (the number of times any request encountered false lock contention) by R744STRC (the total number of lock-related requests), to yield the percent of requests that experienced false lock contention. CPExpert produces Rule WLM652 when this percent is more than the value specified for the **FALSECNT** guidance variable.

The default value for the **FALSECNT** guidance variable is 0.5%, indicating that CPExpert should produce Rule WLM652 when more than one-half of one percent of the lock-related requests encountered false lock contention. You can alter this analysis using the FALSECNT guidance variable. For example, if you wish to be notified when 1 percent of the requests experienced false lock contention, specify:

```
%LET FALSECNT  = 1 ; * ACCEPTABLE PERCENT FALSE LOCK CONTENTION;
```

Chapter 3.8: SYNCSRV variable and LOCKSRV

Signaling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available.

For non-lock structures, CPExpert compares the synchronous service time (R744SSTM) against the **SYNCSRV** variable in USOURCE(WLMGUIDE). For lock structures, CPExpert compares the synchronous service time (R744SSTM) against the **LOCKSRV** variable in USOURCE(WLMGUIDE). Requests to lock structures are very short, and the service time should be significantly less than the requests to non-lock structures.

CPExpert produces Rule WLM660 when the synchronous service time is greater than the SYNCSRV guidance variable for non-lock structures and produces Rule WLM660 when the synchronous service time is greater than the LOCKSRV guidance variable for lock structures.

The default value for the SYNCSRV variable is 350, indicating that CPExpert should produce Rule WLM660 when synchronous service time is more than 350 microseconds for non-lock structures. The default value for the LOCKSRV variable is 250, indicating that CPExpert should produced Rule WLM660 when synchronous service time is more than 250 microseconds for lock structures.

©Copyright 1996, Computer Management Sciences, Inc.

You can alter this analysis using the SYNCSRV and LOCKSRV guidance variables. For example, if you wish to be notified only when the synchronous service time is greater than 500 microseconds for non-lock structures or greater than 100 for lock structures, specify:

```
  %LET SYNCSRV = 500 ; * ACCEPTABLE SERVICE TIME, SYNCH REQUESTS;
%LET LOCKSRV = 500 ; * ACCEPTABLE SERVICE TIME, LOCK REQUESTS;
```

Chapter 3.9: ASYNCSRV variable

Signaling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available.

CPExpert compares the asynchronous service time (R744ASTM) against the **ASYNCSRV** variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM661 when the asynchronous service time is greater than the ASYNCSRV guidance variable.

The default value for the ASYNCSRV variable is 5000, indicating that CPExpert should produced Rule WLM661 when asynchronous service time is more than 5000 microseconds. You can alter this analysis using the ASYNCSRV guidance variable. For example, if you wish to be notified only when the synchronous service time is greater than 10000 microseconds, specify:

```
%LET ASYNCSRV = 10000 ; * ACCEPTABLE SERVICE TIME, ASYNCH REQUESTS;
```

Chapter 3.10: SYNCCHG variable

Signaling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available.

For synchronous requests, XES will either (1) satisfy the request if a subchannel is available, (2) enter CPU "spin-looping" until a subchannel is available and the request is satisfied, or (3) convert the synchronous request to an asynchronous request if the type of request permits the conversion.

CPExpert computes the percent of synchronous requests changed to asynchronous requests (R744SSTA/R744SSRC). The percent of changed requests is compared against the **SYNCCHG** variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM665 when the percent of changed requests is greater than the SYNCCHG guidance variable.

The default value for the SYNCCHG variable is 10, indicating that CPExpert should produced Rule WLM665 when more than 10% of the synchronous requests were changed to asynchronous requests.

You can alter this analysis using the SYNCCHG guidance variable. For example, if you wish to be notified only when more than 15% of the synchronous requests were changed to asynchronous requests, specify:

Chapter 3.11: Specifying guidance for specific structures

The guidance variables for structures as described above are globally applied during CPExpert's analysis of structure performance. These global guidance variables might not be applicable to some structures, however. Guidance can **optionally** be applied to specific structures.

Guidance for specific structures is accomplished by specifying **%LET STRGUIDE=Y;** in USOURCE(WLMGUIDE), identifying the structure(s) to which the guidance applies, and specifying guidance variables for the structure(s).

Exhibit 3-3 illustrates the portion of CPEXPERT.USOURCE(WLMGUIDE) that contains the analysis guidance variables for specific structures. As illustrated in Exhibit 3-3, guidance for individual structures is specified **inside** the SAS macro comment statements (/* and */). The SAS macro comment statements may not be altered, as they control CPExpert's processing of the USOURCE(WLMGUIDE) member.

The structures are identified by the STRUCTURE statement, which is used to specify the structure name to which the specific guidance applies.

Any number of structures may be defined with appropriate guidance specified for the structures.

Following the STRUCTURE statement are the individual guidance statements for the structure identified. Any of the global structure guidance variables can be specified for the structure.

If a structure guidance variable statement is not present or has a null value, the global default will be used. **NOTE: a null value must be indicated by a semi-colon or a SAS error will result.**

The below example shows that specific guidance for the LOCKCONT has been specified for the PIS1DS_IMSIRLM structure. In this example, 100 was specified for the guidance for the LOCKCONT variable. The effect of the specification is to exclude the PIS1DS_IMSIRLM from lock contention analysis.

The below example also shows that specific guidance for the LOCKSRV has been specified for the PIS1DS_IMSIRLM structure. In this example, 50 was specified for the guidance for the LOCKSRV variable.

Note that the SYNCSRV guidance variable was specified as null (no value was used, but a semi-colon followed the equal sign). Normally, this would mean that the global default would be used for the SYNCSRV guidance variable. In this specific example, the PIS1DS_IMSIRLM is a lock structure and the SYNCSRV guidance would have not be used (the LOCKSRV guidance variable would be used for lock structures).

Also note that the SYNCCHG guidance variable is not present in the below example. This illustrates that variables may be missing, and the global default would be used.

```
%LET STRGUIDE = Y;      * GUIDANCE IS PROVIDED FOR INDIVIDUAL STRUCTURES;
/* SPECIFY GUIDANCE FOR STRUCTURES
STRUCTURE   = PIS1DS_IMSIRLM
LOCKCONT    = 100       * MAXIMUM PERCENT LOCK CONTENTION          ;
SYNCSRV     =;          * ACCEPTABLE SERVICE TIME, SYNCHRONOUS REQ  ;
LOCKSRV     = 50        * ACCEPTABLE SERVICE TIME, LOCK REQUESTS    ;
ASYNCSRV    =;          * ACCEPTABLE SERVICE TIME, ASYNCHRONOUS REQ ;
*/
```

Chapter 4: System Logger Analysis Guidance Variables

The system logger is an MVS component that allows an application to log data from a sysplex. The system logger component resides in its own address space on each system in a sysplex. Applications can log data from one system or from multiple systems across the sysplex.

Applications write log data into a *log stream*. From the MVS view, the log stream is a set of records in time sequence order, merged into a single stream, independent of physical residence of the log stream. The log stream can reside in data space storage, in a staging data set, in a coupling facility, or in a log stream DASD data set. System parameters control the placement and length of log stream.

Applications that use the system logger services include:

- **Logrec**. Logrec log stream is an MVS system logger application that records hardware failures, selected software errors, and selected system conditions across the sysplex.

- **Operations log (OPERLOG)**. OPERLOG is an MVS system logger application that records and merges messages about programs and system functions (the hard copy message set) from each system in a sysplex that activates OPERLOG.

- **CICS Log Manager with CICS/Transaction Server for OS**/390. CICS log manager is a CICS system logger application that replaces the journal control management function.

- **IMS Common Queue Server Log Manager**. IMS common shared queues (CQS) log manager is a system logger application that records the information necessary for CQS to recover structures and restart.

- **APPC/MVS**. APPC/MVS is an MVS system logger application that records events related to protected conversations.

- **RRS (resource recovery services)**. RRS is an MVS system logger application that records events related to protected resources.

One significant advantage of the MVS system logger design is that any other system in a sysplex can recover data in the log stream. This feature prevents data loss in case of failure of one system.

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Version Release 1.3 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

©Copyright 1996, Computer Management Sciences, Inc. Revised: April, 2003

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD_only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is coupling facility list structures.

- In a DASD_only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Additionally, for data integrity there exists duplexed storage, so that if one system or component fails, the log stream can be recovered from the duplexed storage. These concepts differ, depending on whether the log stream is defined for a coupling facility or for DASD-only.

- If the primary storage is defined as a list structure in a coupling facility, the duplexed data can be retained in another coupling facility, or can be retained in *staging data sets*. Staging data sets are used when the coupling facility is in the same CPC, or uses volatile storage.

- If the primary storage is defined as DASD-only, the duplexed data is retained in *staging data sets*.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage[9] being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[10]. For log streams defined as DASD-only, these parameters apply to the log stream staging data set.

Once log stream data has been offloaded, the MVS system logger releases the storage in the list structure or staging data set, so the space can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

---

[9] The controls apply **only** to staging data set usage with DASD-only log streams. With coupling facility log streams, the controls apply to both coupling facility structure usage and staging data set usage if the log stream is duplexed to staging data sets.

[10] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets.

CPExpert's system logger analysis guidance variables allows you to guide CPExpert's analysis logger performance problems. Default thresholds have been established based on information contained in IBM publications and other documents. These defaults may not be suitable for your environment and specific management objectives. If the analysis and reports produced by CPExpert do not meet your needs, alter the guidance to CPExpert. If the guidance is insufficient, please call Computer Management Sciences at (703) 922-7027 so we can make changes to improve CPExpert for you!

Exhibit 2-4 illustrates the USOURCE(WLMGUIDE) variables that provide guidance to the WLM Component as it analyzes system logger performance. This chapter describes these variables, how the variables are used, and how the variables are altered.

```
************************************************************************      ;
*          WLM COMPONENT GUIDANCE VARIABLES                                  ;
************************************************************************      ;
          •
          •
          •
          •
**** SYSTEM LOGGER GUIDANCE VARIABLES       ;
%LET SMFTYP88          = N ;     * TYPE 88 RECORDS AVAILABLE IN MXG?   ;
%LET LGDSFULL          =0;       * ACCEPTABLE LOG STREAM STAGING DATA SET FULL;
%LET LGSHIFTS          =1;       * ACCEPTABLE NUMBER OF LOG STREAM DASD SHIFTS;
%LET PCTINTST          =0;       * PERCENT INTERIM STORAGE NOT EFFECTIVELY USED;
%LET PCTLOCST          =0;       * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED;
%LET STDSHIGH          =0;       * STAGING DATA SET HIGH THRESHOLD HIT;
%LET STFULL90          =0;       * ACCEPTABLE STRUCTURE 90% FULL;
%LET STRC2             =0;       * ACCEPTABLE TIMES STRUCTURE HIGHOFFLOAD HIT;
%LET STRC3             =0;       * ACCEPTABLE TIMES STRUCTURE CRITICAL SPACE HIT
%LET STRFULL           =0;       * ACCEPTABLE LOG STREAM STRUCTURE FULL;

%LET LOGGUIDE          = Y;      * GUIDANCE IS PROVIDED FOR INDIVIDUAL LOG STREAMS;
/* SPECIFY GUIDANCE FOR LOG STREAMS
LOGNAME = log.stream.name1
%LET PCTINTST          =0;       * PERCENT INTERIM STORAGE NOT EFFECTIVELY USED;
%LET PCTLOCST          =0;       * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED;
%LET STFULL90          =0;       * ACCEPTABLE STRUCTURE 90% FULL;
%LET STDSHIGH          =0;       * STAGING DATA SET HIGH THRESHOLD ENCOUNTERED;
*/
************************************************************************    ;
```

DEFAULT VALUES FOR SYSTEM LOGGER ANALYSIS

EXHIBIT 2-4

Chapter 4.1: SMFTYP88 variable

The SMFTYP88 guidance variable is used to tell CPExpert whether you wish to analyze system logger performance. If the SMFTYP88 variable is **N**, CPExpert will **not** analyze system logger performance. If the SMFTYP88 variable is **Y**, CPExpert **will** analyze system logger performance.

System logger analysis applies only with a MXG (or SAS/ITSV) performance data base, as CA-MICS does not provide a SAS file contain the SMF Type 88 variables.

If you have a CA-MICS performance data base, and you also have a licensed copy of MXG installed at your site, you can easily perform system logger analysis by running MXG to place SMF Type 88 variables into the MXG TYPE88 data set. You can then define the MXG TYPE88 data set to CPExpert by placing **%LET TYPE88 =saslibname.TYPE88;** in CPExpert's USOURCE(GENGUIDE) guidance member. CPExpert accesses the TYPE88 data as &TYPE88, so you can have CPExpert analyze system logger performance constraints independent of whether CA-MICS processes the data.

Follow the below steps if you have a CA-MICS performance data base and have a licensed copy of MXG at your site:

- Create a SAS library (if necessary) in which MXG can place the TYPE88 information.
- Run MXG to process your SMF data and create the MXG TYPE88 file. Reference your MXG documentation if you are not familiar with this process.
- Specify **%LET TYPE88 = saslibname.TYPE88;** in USOURCE(GENGUIDE). This will define the SAS library to CPExpert.
- Include a JCL DD statement with the name of saslibname and the data set definition of the SAS library in which MXG has placed the TYPE88 information.
- Specify **%LET SMFTYP88 = Y;** in USOURCE(WLMGUIDE). This will tell CPExpert to analyze system logger information.
- Run the WLMCPE module of the WLM Component as normal to analyze data in your CA-MICS performance data base. CPExpert will detect that you wish to analyze system logger performance, and will use the TYPE88 saslibname.TYPE88 definition to acquire the Type 88 information.

©Copyright 1996, Computer Management Sciences, Inc.

Chapter 4.2: LGDSFULL

For a DASD-only log stream or for a log stream that is duplexed to a staging data set, a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set has reached its capacity before off loading data to secondary storage. Once the staging data set space for a log stream is filled, system logger rejects all write requests until the staging data set log data can be offloaded to DASD log data sets.

This situation can cause the application to wait before it can write more data. Depending on the length of time the application must wait, significant performance degradation would be experienced.

CPExpert compares the SMF88ETF (times a staging data set full was detected) variable in the MXG TYPE88 data set with the **LGDSFULL** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM702 when the SMF88ETF value exceeds the **LGDSFULL** guidance variable.

The default value for the **LGDSFULL** guidance variable is zero, indicating that CPExpert should produce Rule WLM702 when any staging data set full condition was detected. You can alter this analysis using the **LGSDSFULL** guidance variable. For example, if you wish to be notified only when the staging data set for the log stream is full more than 5 times in an SMF interval, specify:

```
%LET LGDSFULL = 5 ; * ACCEPTABLE LOG STREAM STAGING DATA SET FULL;
```

From a practical matter, you should always wish to be notified that a log stream staging data set experienced a “FULL” condition. Please notify Computer Management Sciences at (703) 922-7027 if you have a situation where you wish to modify this guidance variable (so we can better appreciate unique situations).

Chapter 4.3: LGSHIFTS

A log stream can have data in multiple DASD log data sets. As an offload data set becomes full, the system logger automatically allocates a new one for the log stream. This process is known as a “DASD-shift” and *generates considerable overhead*. Consequently, a “DASD-shift” should not occur frequently. IBM suggests that “DASD-shifts” should occur no more than once per hour.

CPExpert examines the SMF88EDS variable (the number of log stream DASD shifts during the SMF interval). While IBM suggests that you not have more than one DASD shift per hour, an SMF recording interval typically is less than an hour (normally the interval is 15

minutes). Consequently, CPExpert calculates the number of SMF intervals in an hour and tracks the number of DASD shifts that occur during any hour.

CPExpert produces Rule WLM707 when the number of DASD shifts that occur during any hour exceeds the **LGSHIFTS** guidance variable in USOURCE(WLMGUIDE). The default value for the **LGSHIFTS** is one, indicating that CPExpert should produce Rule WLM707 when more than one log stream DASD shift occurred during any hour. You can alter this analysis using the **LGSDSFULL** guidance variable. For example, if you wish to be notified only when more than two DASD shifts occur in one hour, specify:

```
%LET LGSHIFTS = 2 ; * ACCEPTABLE NUMBER OF LOG STREAM DASD SHIFTS;
```

Chapter 4.4: PCTINTST

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[11], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

Once log stream data has been offloaded, the MVS system logger releases the storage in the list structure, so the space in the structure can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

[11] Interim storage is sometimes referred to as "primary" storage.

CPExpert computes the percent of ineffective use of interim storage (PCTINTST) by applying the following algorithm:

$$PCTINTST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where:

SMF88SAB = Bytes deleted after being offloaded
SMF88SIB = Bytes deleted before being offloaded

CPExpert compares the computed PCTINTST with the **PCTINTST** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM704 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTINTST** guidance variable.

The default value for the **PCTINTST** guidance variable is zero, indicating that CPExpert should produce Rule WLM704 whenever interim storage use was not effective. You can alter this analysis using the **PCTINTST** guidance variable. For example, if you wish to be notified only when more than ten percent of the bytes were deleted after offload, specify:

```
%LET PCTINTST = 10% ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USED;
```

Chapter 4.5: PCTLOCT

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[12], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is “hardened” for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium that the system logger uses to hold interim log data:

- With a coupling facility log stream, interim storage is contained in coupling facility list structures. The system logger duplexes the log stream to either (1) MVS data space areas associated with the system logger address space or (2) staging data sets, depending on whether the coupling facility is failure-independent. Interim storage

[12] Interim storage is sometimes referred to as “primary” storage.

residing in coupling facility structures is analyzed by Rule WLM704 and the analysis is guided by the PCTINTST guidance variable.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to staging data sets. Interim storage residing in local storage buffers is analyzed by Rule WLM705 and the analysis is guided by the PCTLOCST guidance variable.

Interim storage normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[13]. For log streams defined as DASD-only, these parameters apply to the **log stream staging data set**.

Once log stream data has been offloaded, the MVS system logger releases the storage in the staging data sets, so the space in the staging data sets can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

CPExpert computes the percent of ineffective use of staging data sets (PCTLOCST) by applying the following algorithm to DASD-only log streams

$$PCTLOCST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where

SMF88SAB = Bytes deleted after being offloaded

SMF88SIB = Bytes deleted before being offloaded

[13] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets. Problems with staging data set threshold being encountered are analyzed in Rule WLM705.

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM705 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTLOCST** guidance variable.

The default value for the **PCTLOCST** guidance variable is 0, indicating that CPExpert should produce Rule WLM705 whenever DASD staging data set use was not effective. You can alter this analysis using the **PCTLOCST** guidance variable. For example, if you wish to be notified only when more than ten percent of the local storage bytes were deleted after offload, specify:

```
%LET PCTLOCST = 10% ; * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED;
```

Chapter 4.6: STDSHIGH

Interim storage in a coupling facility structure normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

Additionally, for data integrity there exists duplexed storage, so that if one system or component fails, the log stream can be recovered from the duplexed storage. These concepts differ, depending on whether the log stream is defined for a coupling facility or for DASD-only.

- If the primary storage is defined as a list structure in a coupling facility, the duplexed data can be retained in another coupling facility, or can be retained in *staging data sets*. Staging data sets are used when the coupling facility is in the same CPC, or uses volatile storage.

- If the primary storage is defined as DASD-only, the duplexed data is retained in *staging data sets*.

When a log stream in a coupling facility is duplexed to staging data sets, the system logger automatically makes a duplicate copy of the data every time data is written to a log stream. This is done to protect against data loss due to coupling facility problems or due to system failure. The duplicate copy is kept in the staging data sets until the data is off-loaded from the coupling facility structure to DASD log data sets. After the data is off-loaded to DASD log data sets, the system logger discards the duplicate copy of the log data.

Interim storage in a coupling facility structure normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and

LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled. For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the coupling facility to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **low offload threshold** is the point in the coupling facility structure, as a percent space consumed , where the system logger stops offloading coupling facility log data to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

For log streams in a coupling facility that are duplexed to staging data sets, the values of the HIGHOFFLOAD and LOWOFFLOAD parameters **apply to the staging data sets** as well as to the coupling facility structure. This is simply because if the staging data sets become full, MVS would not be able to continue duplexing data and there would be a data integrity exposure in case of failure. Consequently, if a staging data set fills up **before** an offload of a log stream in a coupling facility structure is triggered by the high threshold specification, an offload will be triggered because of the full staging data set.

When a staging data set reaches the high threshold, the system logger immediately offloads data from the coupling facility to DASD log data sets, even if the coupling facility usage for the log stream is below the high threshold. Thus, if the staging data sets are small in comparison to the coupling facility structure size for a log stream, the staging data sets will keep filling up and the system logger will frequently offload coupling facility data to DASD log data sets. This means that your installation would experiences frequent (and unexpected) offloading overhead that could affect performance[14].

CPExpert examines the SMF88STN variable (the structure name) in the MXG TYPE88 data set to select records that apply only to coupling facility structures[15]. For these records, CPExpert examines the SMF88ETF variable (the number of times the system logger detected a Staging Data Set Threshold hit condition). CPExpert produces Rule WLM706 when the SMF88ETF value exceeds the **STDSHIGH** guidance variable in USOURCE(WLMGUIDE).

---

[14] If your staging data sets are too small, you also run the risk of filling them up completely. If this occurs, system logger immediately begins offloading the coupling facility log data in DASD log data sets to harden it. System logger applications will be unable to log data until system logger can free up staging data set space. This serious situation is evaluated by Rule WLM702.

[15] The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

The default value for the **STDSHIGH** is zero, indicating that CPExpert should produce Rule WLM706 whenever a Staging Data Set Threshold was encountered. You can alter this analysis using the **STDSHIGH** guidance variable. For example, if you wish to be notified only when the HIGHOFFLOAD threshold was encountered more than 5 times during an SMF recording interval, specify:

```
%LET STDSHIGH = 5 ; * TIMES STAGING DATA SET HIGH THRESHOLD HIT;
```

Chapter 4.7: STFULL90

When a coupling facility structure is defined, it is divided into two areas: One area holds *list elements*, and the other area holds *list entries*. List elements are units of logged data and are either 256 bytes or 512 bytes long. There is at least one element per log record. List entries are index pointers to the list elements. There is one list entry per log record.

Each log record places an entry in the list entry area of the structure, and the data is loaded as one or more elements in the list element area. **If the list entry area exceeds 90% of its capacity, all log streams are offloaded to DASD.** DASD offloading commences at this point, regardless of the current utilization of the log stream, and continues until an amount of data equal to the difference between the HIGHOFFLOAD threshold and the LOWOFFLOAD threshold for the log stream has been offloaded.

This situation can occur when log steams share a structure, one log stream is used by an application issuing very few journal write requests, and other applications issue frequent journal write requests to log streams in the same structure. All log streams may be offloaded to DASD because of the frequent journal write requests by the other applications.

The primary disadvantage of encountering this situation is that the application that is infrequently writing to the log stream might not have its LOWOFFLOAD and HIGHOFFLOAD thresholds controlling the offload process. This can result in unpredictable offloading, and possibly undesirable performance.

For example, Log Stream A might have a HIGHOFFLOAD threshold of 80% and a LOWOFFLOAD threshold of 60%. Because of log stream activity by other applications writing to other log streams, the list entry area may exceed 90% of its capacity even though Log Stream A might be only 50% utilized. Although Log Stream A had not reached its HIGHOFFLOAD threshold, or even its LOWOFFLOAD threshold, data would be offloaded until 20% of the log stream was offloaded. This is the difference between 80% and 60%. After the offloading operation has completed, log stream A is at 30% utilization (50% minus 20%).

©Copyright 1996, Computer Management Sciences, Inc.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88STN variable (the structure name) to select information that applies only to coupling facility structures[16]. For these records, CPExpert examines the SMF88EFS variable (offloads for all log streams connected from this system to this structure because structure was 90% full) in the SMF Type 88 records. CPExpert produces Rule WLM703 when the SMF88EFS value exceeds the **STFULL90** guidance variable in USOURCE(CICGUIDE).

The default value for the **STFULL90** is zero. Any non-zero value in the SMF88EFS variable indicates that the entry to element ratio is too high for the structure. You can alter this analysis using the **STFULL90** guidance variable. For example, if you wish to be notified only when offloads were scheduled because the structure was 90% full more than 5 times during an SMF recording interval, specify:

```
%LET STFULL90 = 5 ; * TIMES OFFLOADS BECAUSE STRUCTURE WAS 90% FULL;
```

Chapter 4.8: STRC2

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility

[16] The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

structure to DASD log data sets or if the system logger configuration is tuned incorrectly. The Type-3 completions are analyzed by Rule WLM309.

CPExpert examines the SMF88SC2 variable (Count of Type-2 completions) in the SMF Type 88 records. CPExpert produces Rule WLM708 when the SMF88SC2 value exceeds the **STRC2** guidance variable in USOURCE(WLMGUIDE).

The default value for the **STRC2** is zero, indicating that CPExpert should produce Rule WLM708 whenever the HIGHOFFLOAD threshold was reached in an SMF interval. You can alter this analysis using the **STRC2** guidance variable. For example, if you wish to be notified only when the structure reached the HIGHOFFLOAD threshold more 5 times during an SMF recording interval, specify:

```
%LET STRC2 = 5 ; * TIMES STRUCTURE HIGHOFFLOAD THRESHOLD WAS HIT;
```

Chapter 4.9: STRC3

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets. The Type-2 completions are analyzed by Rule WLM308.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility structure to DASD log data sets or if the system logger configuration is tuned incorrectly. If a log stream has a large proportion of Type-3 completions, the system logger is getting dangerously close to the STRUCTURE FULL condition.

CPExpert examines the SMF88SC3 variable (Count of Type-3 completions) in the SMF Type 88 records. CPExpert produces Rule WLM709 when the SMF88SC3 value exceeds the **STRC3** guidance variable in USOURCE(WLMGUIDE).

The default value for the **STRC3** is zero, indicating that CPExpert should produce Rule WLM709 whenever the space used by a log stream caused the coupling facility structure to reach a critical amount. You can alter this analysis using the **STRC3** guidance variable. For example, if you wish to be notified only when the log stream caused the structure to reach a critical amount more than 5 times during an SMF recording interval, specify:

```
%LET STRC3 = 5 ; * TIMES STRUCTURE CRITICAL SPACE  WAS HIT;
```

Chapter 4.10: STRFULL

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Version Release 1.3 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

- For a log stream that uses a coupling facility structure, a 'STRUCTURE FULL' condition can exist. In this case, the coupling facility has reached its capacity before off loading data to DASD[17]. This condition is analyzed by Rule WLM701.

- For a DASD-only log stream or for a log stream that is duplexed to a staging data set, a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set has reached its capacity before off loading data to secondary storage. This condition is analyzed by Rule WLM702.

If either of the above situations occur, they indicate that the logger cannot write data to secondary storage quickly enough to keep up with incoming data. Once the coupling facility space for a log stream is filled, system logger rejects all write requests until the coupling facility log data can be offloaded to DASD log data sets. Both situations can cause the application to wait before it can write more data. Depending on the length of time the application must wait, significant performance degradation would be experienced.

CPExpert examines the SMF88STN variable in the MXG TYPE88 data set (this variable indicates whether the log stream is a coupling facility type, or is a DASDONLY type). When this variable indicates the log stream is a coupling facility type, CPExpert compares the SMF88ESF (times a structure full condition was detected) variable in the MXG

[17] This condition could be encountered during the rebuilding of a coupling facility structure, but rebuilding of a coupling facility structure is an event that would not require CPExpert's analysis - such an event would be well-known to systems personnel!

TYPE88 data set with the **STRFULL** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM701 when the SMF88ESF value exceeds the **STRFULL** guidance variable.

The default value for the **STRFULL** guidance variable is zero, indicating that CPExpert should produce Rule WLM701 when any structure full condition was detected. You can alter this analysis using the **STRFULL** guidance variable. For example, if you wish to be notified only when a structure full condition was encountered more than 5 times during an SMF recording interval, specify:

```
%LET STRFULL = 5 ; * TIMES STRUCTURE FULL WAS ENCOUNTERED;
```

Chapter 4.11: Specifying guidance for specific log streams

The guidance variables for log streams as described above are globally applied during CPExpert's analysis of system logger performance. These global guidance variables might not be applicable to some log streams, however. Guidance can **optionally** be applied to specific log streams <u>or to coupling facility structures used by the log streams</u>.

Guidance for specific log streams (or log stream structures) is accomplished by specifying **%LET LOGGUIDE=Y;** in USOURCE(WLMGUIDE), identifying the log stream(s) or structure(s) to which the guidance applies, and specifying guidance variables for the log stream(s) or structrure(s).

Exhibit 3-5 illustrates the portion of CPEXPERT.USOURCE(WLMGUIDE) that contains the analysis guidance variables for specific log streams and log stream structures. As illustrated in Exhibit 3-5, guidance for individual log streams or log stream structures is specified **inside** the SAS macro comment statements (/* and */). The SAS macro comment statements may not be altered, as they control CPExpert's processing of the USOURCE(WLMGUIDE) member.

The log streams or log stream structures are identified by the LOGNAME statement, which is used to specify the log stream name or log stream structure name to which the specific guidance applies.

Any number of log streams or log stream structures may be defined with appropriate guidance specified for the log streams or structures.

Following the LOGNAME statement are the individual guidance statements for the log stream or structure identified. Any of the global log stream guidance variables can be specified for the log stream or structure.

If a log stream guidance variable statement is not present or has a null value, the global default will be used. **NOTE: a null value must be indicated by a semi-colon or a SAS error will result.**

The below example shows that specific guidance for the has been specified for the SYSPLEX.OPERLOG log stream. In this example, 9999999 was specified for the guidance for the STRC2 variable. The effect of the specification is to exclude the SYSPLEX.OPERLOG from HIGHOFFLOAD analysis (as described in Rule WLM708).

```
%LET LOGGUIDE = Y;      * GUIDANCE IS PROVIDED FOR INDIVIDUAL LOG STREAMS;
/* SPECIFY GUIDANCE FOR LOG STREAMS
LOGNAME  = SYSPLEX.OPERLOG
STRC2       = 99999999    *ACCEPTABLE TIMES STRUCTURE HIGHOFFLOAD HIT          ;
*/
```

Section 3: Using the WLM Component

This section describes how to use the WLM Component to analyze system performance, to identify intervals when service classes did not achieve their performance goals, and to determine why the performance goals were not met.

Before executing the WLM Component, you should review the guidance contained in the prefix.CPEXPERT.USOURCE(WLMGUIDE) PDS member, to make sure that the guidance variables are appropriate for your environment. Exhibit 2-1 illustrates the variables that must be changed in the WLMGUIDE PDS member.

Chapter 1: Executing the WLMCPE Module

This chapter describes how to execute the WLMCPE Module of the WLM Component. We suggest that you execute the WLMCPE Module on a daily basis, after the normal update of your performance data base.

As stated in the Introduction to this document, the WLM Component consists of numerous modules, working together to (1) detect intervals in which service classes did not achieve performance goals, (2) shape system performance and utilization data for detailed analysis by other modules, (3) evaluate the data to assess reasons service classes did not achieve performance goals, and (4) report the results from the evaluation. Additionally, the WLM Component contains modules to analyze the Workload Manager service definition to detect potential problems with workload classification or service class definitions. These modules are loaded and controlled by the central WLM Component of CPExpert (titled WLMCPE).

Step 1. Use TSO ISPF to change the "prefix" in the data set names

Use TSO ISPF to change the "prefix" in the data set names (DSN) in the **USOURCE** DD statement, the **SOURCE** DD statement, the **CPEDATA** DD statement, the **HISTORY** DD statement, the **PDBLIB** DD statement, and the **SYSIN** DD statement of the JCL in accordance with your installation standards. The JCL is illustrated in Exhibit 3-1. (A "shell" of this JCL is contained on the distribution tape as "WLMJCL1")

- The CPEDATA DD statement in Exhibit 3-1 refers to the SAS data library maintained by CPExpert. The space for this library was created during the installation of CPExpert.

```
//jobname     JOB    job card information
//STEP01      EXEC   SAS,OPTIONS='MACRO DQUOTE PAGESIZE=65 ERRORABEND'
//USOURCE     DD     DSN=prefix.CPEXPERT.USOURCE,DISP=SHR
//SOURCE      DD     DSN=prefix.CPEXPERT.SOURCE,DISP=SHR
//CPEDATA     DD     DSN=prefix.CPEXPERT.CPEDATA,DISP=OLD
//HISTORY     DD     DSN=prefix.CPEXPERT.HISTORY,DISP=OLD
//PDBLIB      DD     DSN=prefix.MXG.PDBLIB.SASDB,DISP=SHR
//LIBRARY     DD     DSN=saslib containing MXG FORMATs
//CPEDASD     DD     -optional-
//SYSIN       DD     DSN=prefix.CPEXPERT.SOURCE(WLMCPE),DISP=SHR
```

JOB CONTROL LANGUAGE TO EXECUTE THE WLMCPE MODULE

EXHIBIT 3-1

- The HISTORY DD statement in Exhibit 3-1 refers to the SAS history data library maintained by CPExpert. The space for this library was created during the installation of CPExpert.

- The PDBLIB DD statement in Exhibit 3-1 refers to the SAS library containing the performance data base to be analyzed. The example shows a sample DSN for a typical MXG performance data base. The DSN would be changed to "DSN=prefix.RMF.MICS.DETAIL" to use a MICS performance data base.

  Exhibit 3-1 does not show the optional DD statements for MICS Information Areas (i.e., BATLIB DD, SCBLIB DD, HARLIB DD, etc.). The CPExpert Installation Guide describes how to use these optional DD statements if you have your MICS performance data base separated by MICS Information Area.

- The SASLIB DD statement in Exhibit 3-1 refers to the SAS library containing the MXG FORMATs. If you execute CPExpert against a MICS performance data base, the SASLIB DD statement would be changed to refer to the MICS FORMAT library.

- The **optional** CPEDASD DD statement in Exhibit 3-1 applies only if you have licensed the DASD Component of CPExpert and you have installed the modification to MXG or MICS which collects SMF Type 30 (Data Definition) information.

  The WLM Component estimates potential delays to service classes, including delays caused by DASD. If you have not licensed the DASD Component, the WLM

Component uses the average DASD I/O service time for all devices to assess potential delays to TSO. If you have licensed the DASD Component and have installed the modification to MXG or MICS, the WLM Component can determine which DASD devices are used by service classes and thus can provide a better estimate of potential problems related to DASD service delays.

Step 2: Make any appropriate changes to the WLMGUIDE Module

Before submitting the above JCL shown in Exhibit 3-1 and executing the WLMCPE Module, you should make appropriate changes to the prefix.CPEXPERT.USOURCE(WLMGUIDE) module. These changes are described in Section 2 of this manual.

The defaults provided with the WLM Component should be adequate for initial executions of the WLM Component. You normally will make changes to the guidance only after executing the WLM Component several times.

Step 3: Review the output from the WLMCPE Module

If any rules produced, refer to the specific rule in Appendix A of this document for a description of the rule, a discussion of why the rule produced, and a suggestion for actions that should be taken.

Depending upon the output, you may wish to make changes or wait to see if the problems are identified in an analysis of a subsequent day's performance.

The following points should be considered in deciding whether to make a change:

- The WLM Component may identify problems which clearly should be solved because their effect is so serious. In many cases, once the problem is identified, users immediately realize that the problem and suggested solutions make sense.

- The WLM Component may identify problems which you do not feel will commonly occur. For example, you may realize that the results are based upon abnormal workload and changes may be unnecessary since the conditions will not occur often.

- **You generally should make only one change at a time if you decide to make changes!** This sound tuning advice if founded on the principles that:

  - Tuning is an art. No one (and certainly not CPExpert) can guarantee that any particular change will have a beneficial effect in all environments.

- Changes may have unexpected effects. Most systems are complex, parameters may improve performance of one area at the expense of performance in another area, and management may wish resources focused on the second area.

- If you make multiple changes and performance deteriorates, you will be unable to identify easily the change causing the problem. You are then faced with the problem of backing out all of the changes and starting over, one at a time.

- Some changes are not "precise" in that, for example, keyword values might need to be adjusted a little at a time until a suitable value is reached. If multiple changes are made, you will be unable to detect the effect of the fine- tuning of the changes.

- Above all, **remember that the recommendations from CPExpert are simply options** to be considered in the context of overall objectives. You must decide whether the recommendations are reasonable. Rarely should a recommendation be implemented without first evaluating how the recommendation will effect other workloads.

  Please remember that CPExpert is not intended to replace a performance analyst. Rather, CPExpert was developed to help analyze the performance of MVS systems. CPExpert automates much of the routine of computer performance evaluation. Performance analysts can then focus on the areas which are not routine and which "require thinking".

  With this philosophy, please let us know when you discover areas in which CPExpert could have helped you analyze a problem. We will improve our product and you will have more help!

Chapter 2: Checklist

The following checklist should be followed to execute the WLM Component of CPExpert.

Checklist for Executing the WLM Component, Mainframe

- ☐ Change the job card information in the JCL.
- ☐ Change the "prefix" in the data set names in the DD statements.
  - ☐ Change the "prefix" in USOURCE DD statement.
  - ☐ Change the "prefix" in SOURCE DD statement.
  - ☐ Change the "prefix" in CPEDATA DD statement.
  - ☐ Change the "prefix" in HISTORY DD statement.
  - ☐ Change the "prefix" in PDBLIB DD statement.
  - ☐ Change the "prefix" in SASLIB DD statement.
  - ☐ Optional: Change the "prefix" in CPEDASD DD statement. This step is required only if you have licensed the DASD Component of CPExpert and have installed the modification to MXG or MICS to collect SMF Type 30 DASD information. Please refer to the DASD Component User Manual for further information.
- ☐ Make any necessary changes to the WLMGUIDE Module in USOURCE.
- ☐ Submit the JCL to execute the WLMCPE Module.

©Copyright 1994, Computer Management Sciences, Inc.

Section 4: Analyzing the Workload Manager

This section discusses considerations for analyzing the Workload Manager.

- Chapter 1 presents an overview of the Workload Manager concepts.
- Chapter 2 describes subsystem transactions.
- Chapter 3 discusses internal logic of the Workload Manager.
- Chapter 4 describes the performance data which are used by CPExpert to analyze system performance from the perspective of the Workload Manager.
- Chapter 5 highlights some of the factors that must be considered when analyzing performance based upon workload data collected and recorded by RMF.

The purpose of this section is to present information which may not be generally available. The basic IBM documents dealing with the Workload Manager (the *Planning: Workload Management* document and the *Programming: Workload Management Services* document) describe the basic structure of the Workload Manager and user interaction with the Workload Manager. Some of the concepts are explained well in these documents, while other concepts have not been completely addressed.

IBM Workload Manager developers have given presentations at various professional conferences to amplify the Workload Manager concepts. This section (1) presents some of the discussion from the conferences, (2) describes concepts based on our analysis of SMF data from systems executing under SP5 (Goal Mode), and (3) elaborates concepts based on private discussions with the IBM Workload Manager developers.

Not all concepts are completely understood, and this section will be improved with increased knowledge. More importantly, the results from Workload Manager parameter specification in different environments can be quite different from the results the Workload Manager developers envisioned. This particularly is true of subsystem (e.g., CICS Version 4.1) interaction with the Workload Manager. The WLM Component of CPExpert will be improved (including this section) as SP5 (Goal Mode) is implemented in more installations and more information becomes available.

As you use the WLM Component and gain additional experience with the Workload Manager, please call Computer Management Sciences with any information which might be different from that presented in this section! For that matter, feel free to call with any questions or comments. **We appreciate feed-back**.

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 1: Workload Manager Concepts

As mentioned in Section 1, the Workload Manager introduced with MVS/ESA SP5.1 is a radical departure from earlier system management concepts employed with MVS. The Workload Manager attempts to solve problems with earlier versions of MVS by removing the requirement that users provide detailed guidance to MVS on how MVS should process work. With the Workload Manager, users specify **performance goals**. The Workload Manager interprets the performance goals, and the Workload Manager tells MVS how to process work to meet the performance goals.

Some of the interaction between users and MVS is similar to earlier systems.

- Users still categorize work using workload categorization schemes similar to the schemes embodied in the IEAICSxx definitions.

- The workload is assigned to *service classes*, much as workload was assigned to performance groups in earlier versions of MVS.

The major difference between earlier versions of MVS and the Workload Manager is that users do not provide the Workload Manager with detailed specifications on how to process the work assigned to a service class. Users describe **performance goals**, and the Workload Manager adjusts system resources to meet the performance goals.

Chapter 1.1: Service Definition

Users define workload categories, establish classification rules for the workloads, assign the workloads to service classes, define performance goals and goal importance for the service classes, and establish resource boundaries for the service classes by constructing a *service definition*. Users construct the service definition using the Workload Manager ISPF application. A service definition is simply a collection of the output from the Workload Manager ISPF application. There is one service definition for an entire sysplex. The service definition consists of *service policies*, *workload classification schemes* for the workload, *service classes*, and *resource groups*. Each of these components of a service definition will be described in the following chapters.

Chapter 1.2: Service Policies

Users communicate their service requirements for workloads by defining a *Service Policy* to the Workload Manager. A service policy is simply a "named" collection of service classes and their associated performance goals and (optionally) processing boundaries for the address spaces associated with the service classes. Each of these parts of the service policy will be described in the following chapters.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

A policy applies to a sysplex. Once a policy is invoked (using an operator command), the definitions contained in the policy remain in effect for the duration of the IPL. Although part or all of the policy may be overridden by a different policy invoked by an operator, any definition not overridden will remain in effect.

Chapter 1.3: Workload Classification

In earlier versions of MVS, installations classified work using the IEAICSxx member of SYS1.PARMLIB. After applying classification rules in IEAICSxx, the resulting work would be assigned to performance groups.

With the Workload Manager, the same process conceptually applies. However, instead of assigning work to performance groups, work is classified after applying classification rules and the workload is assigned to a service class[1].

Work is classified based on "qualifiers" associated with the work[2]. Exhibit 4-1 shows the work qualifiers available with each version of MVS.

- Accounting information (AI): accounting information provided with the job or transaction.
- Correlation information (CI): correlation information is the DB2 correlation ID of the DDF server thread.
- Collection name (CN): collection name is the DB2 collection name of the first SQL package accessed by the distributed relational database architecture (DRDA) requestor in the work request.
- Connection type (CT): connection type is the DB2 connection type of the distributed data facility (DDF) server thread. The thread contains the value "DIST ' indicating it is a server.
- LU Name (LU): the logical unit (LU) name originating the work.
- Netid (NET): the network identification originating the work.
- Process Name (PC): the MQWIH_ServiceName from the message's work information header.

[1]The workload may be assigned to a report class instead of being assigned to a service class or in addition to being assigned to a service class. Report classes are described in Chapter 1.5.

[2]See IBM's *Planning: Workload Management* document (Chapter 10: Defining Classification Rules) for a complete discussion of the workload classification rules.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

- Perform (PF): performance group number specified via PERFORM keyword on the JCL JOB statement or the START command, of the performance group number specified on the TSO logon panel.

- Package name (PK): package name is the name of the first DB2 package accessed by the DRDA requestor in the work request.

- Plan name (PN): plan name is the DB2 plan name associated with the DB2 server thread.

- Procedure Name (PR): the procedure name associated with the originator of the query.

- Priority (PRI): the JES priority associated with the batch job submitted through JES2.

- Sysplex name (PX): the sysplex name of the specific sysplex in which the work is running.

- Scheduling environment (SE): for JES, this is the scheduling environment name assigned to the job; for DB2, this is the scheduling environment name associated with the originator of the query.

- Subsystem instance (SI): the subsystem which generated the work. IBM-supplied subsystems consist of ASCH (APPC transaction programs), CICS (transactions processed by CICS), DDF (work requests from DB2 distributed data facility), IMS (messages processed by IMS), and JES (either JES2 or JES3).

- Subsystem specific parameter (SPM): a parameter that a subsystem can define and use to further classify work.

- Subsystem collection name (SSC): for JES, this is the XCF group name; for DB2, this is the subsystem collection name associated with the originator of the query.

- System name (SY): the system name for those address spaces whose execution system is known at classification time. This applies to ASCH, OMVS, STC, SYSH[3], and TSO. JES is not eligible for this qualifier, as the system on which classification occurs might not be the system on which the job is run. Additionally, subsystem-defined transactions (CICS/IMS) and enclave-based transactions are not bound to an execution system at classification time, and are not eligible for the system name qualifier.

- Transaction class/job class (TC): the transaction class or job class.

- Transaction name/job name (TN): the transaction name or job name.

[3] The SYSH address space was introduced with z/OS V1R2 to describe logical partitions running LINUX or other non z/OS systems. CPU resources can be managed across these logical partitions in accordance with workload goals.

- Userid (UI): the user identification associated with the job, transaction, message, etc.

Unlike the classification scheme used with MVS prior to SP5.1 (Goal Mode), there is no set order of the classification. Users of the Workload Manager define the order in which the work qualifiers are to be applied to work arriving in the system.

MVS assigns defaults if work coming into the system is not classified by the classification scheme defined by users. The defaults are based on the type of work:

- SYSTEM (consisting of system address spaces): MASTER, GRS, DUMPSRV, SMF, CATALOG, RASP, XCFAS, SMXC, CONSOLE, IOSAS, and all other system address spaces designated "high dispatching priority" by MVS (that is, those with a dispatching priority of X'FF').

- SYSSTC (consisting of started tasks not classified by the workload classification scheme).

- SYSOTHER (for all other work not classified by the workload classification scheme).

©Copyright 1994, Computer Management Sciences, Inc.

		OS/390									
QUALIFIER	**SP5**	**V1R1**	**V1R2**	**V1R3**	**V2R4**	**V2R5**	**V2R6**	**V2R7**	**V2R8**	**V2R9**	**V2R10**
AI	X	X	X	X	X	X	X	X	X	X	X
CI		X	X	X	X	X	X	X	X	X	X
CN		X	X	X	X	X	X	X	X	X	X
CT		X	X	X	X	X	X	X	X	X	X
LU	X	X	X	X	X	X	X	X	X	X	X
NET	X	X	X	X	X	X	X	X	X	X	X
PC							X	X	X	X	X
PF				X	X	X	X	X	X	X	X
PK		X	X	X	X	X	X	X	X	X	X
PN		X	X	X	X	X	X	X	X	X	X
PR				X	X	X	X	X	X	X	X
PRI					X	X	X	X	X	X	X
PX											X
SE											X
SI	X	X	X	X	X	X	X	X	X	X	X
SPM				X	X	X	X	X	X	X	X
SSC											X
SY											X
TC	X	X	X	X	X	X	X	X	X	X	X
TN	X	X	X	X	X	X	X	X	X	X	X
UI	X	X	X	X	X	X	X	X		X	X

WORKLOAD CLASSIFICATION QUALIFIERS AVAILABLE WITH EACH VERSION OF MVS

EXHIBIT 4-1

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 1.4: Service Classes

Users assign workload to a service class and specify the performance goal and goal importance of the service class. Optionally, users also may (1) define multiple performance periods for the service class and (2) assign the service class to a resource group.

A service class can represent any collection of workload that can be classified using the workload classification schemes available with the Workload Manager.

Exhibit 4-2 shows example service classes and their associated workloads.

SERVICE CLASS	WORKLOAD DESCRIPTION
TSOPROD	TSO transactions submitted by production organizations
TSOTEST	TSO transactions submitted by test and development organizations
BATCH	Regular batch jobs
LOWPRI	Low priority batch jobs
CICSAOR1	CICS Application Owning Region 1
CICSPERS	CICS transactions from Personnel Dept
CICSADMN	CICS transactions from Administration Dept
CICSOPS	CICS transactions from Operations Dept
ST_TASK	Started tasks

EXAMPLE SERVICE CLASSES

EXHIBIT 4-2

The example shows that service classes may be similar to performance groups or domains which were defined with earlier versions of MVS.

- Service classes are much like performance groups, in that they have workloads assigned to them. As will be clear from later discussion, address spaces associated

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

with a service class may migrate to lower service class periods based upon the duration of service used by the address space. Since the Workload Manager manages at the *service class period* level, subsequent discussion generally will refer to service class *periods* rather than referring to service classes.

- Service class periods also are like domains from the view of the SRM in that SRM controls in such areas as the target MPL or expanded storage controls may be applied to service class periods. In fact, the Workload Manager creates a domain (for the SRM's internal control tables) for each service class period defined in a service policy.

There are some significant differences between (1) performance group periods and domains, and (2) service class periods:

- With performance group periods, users can specify detailed controls (e.g., dispatching priority, timeslicing, storage isolation, response time option, etc.). With domains, users similarly can specify detailed controls (i.e., minimum and maximum multiprogramming levels, service slopes, and expanded storage criteria).

- With service class periods, users can specify performance goals and importance of the goal. Additionally, users can assign the service class period to a resource group (described later) and specify resource minimum and maximum for the resource group.

- If CICS/ESA Version 4.1 or IMS Version 5 are installed, users can specify service classes *at the transaction level* for transactions which are served by a CICS region or IMS region. These subsystem transactions can be grouped into service classes, and the Workload Manager will attempt to achieve transaction response based on the performance goals specified for the transaction service classes.

  Exhibit 4-2 shows three transaction service classes (CICSPERS, CICSOPS, and CICSADMN). These transaction service classes are considered to be "served" transaction service classes, and they are served by the CICSAOR1 "server" service class in the example.

  There are several significant restrictions with assigning transactions to service classes:

  - The service class describing the transactions must have a response performance goal and importance assigned; however, the service class cannot have multiple performance periods. This is because the SRM does not collect resource use at the transaction level. Rather, all resource use is accounted for at the subsystem level (i.e., at the CICS/ESA Version 4.1 region or IMS Version 5 region level).

    Since resource information is not collected at the transaction level, "service units" cannot be attributed to transactions. Consequently, there is no way for the SRM to determine how much "service" the transactions have accumulated. Since the SRM

cannot determine how much service the transactions have accumulated, the Workload Manager cannot migrate transactions to lower performance periods.

Further, CICS or IMS actually handle dispatching of the transactions within the CICS or IMS region. Consequently, the Workload Manager could not take action at the transaction level even if it did determine how much service the transactions accumulated. The Workload Manager can manage resources only at the CICS or IMS region level, as these are the address spaces actually using resources.

- The subsystem service class (e.g., the CICS/ESA Version 4.1 region or IMS Version 5 region) must have a performance goal and importance defined, in order for the region to start-up. However, the performance goal and importance normally are used by the Workload Manager **only at start-up time** for the address space[4].

After start-up time, the Workload Manger normally ignores the goal and importance of subsystems. After start-up time, the Workload Manager normally uses the goal and importance of the "served" transaction service classes as the basis for it resource allocation decisions.

The Workload Manager attempts to meet the performance goals of the "served" transaction service classes. In order to meet these performance goals, the Workload Manager must assign resources to the server service class (e.g., the service class of the CICS region), regardless of the goal and importance assigned to the subsystem service class.

Chapter 1.4.1: Performance goals

Users specify a performance goal for each service class. There are four types of performance goals: average response, percentile response, execution velocity, and discretionary.

- **Average response goal**. For service classes which have an average response time goal, the Workload Manager determines whether the average response time achieved by transactions ending in the service class is greater than the performance goal. If the average response time is greater than the performance goal, the system is not meeting performance goals for the service class period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

[4]This statement is not true if the region should become idle for some period of time. If there are no transactions executing in the region for some time, the Workload Manager will rely on the performance goal and importance associated with the region to make resource allocation decisions. This situation should normally occur only during "off shifts" or for test regions with low activity.

©Copyright 1994, Computer Management Sciences, Inc.

- **Percentile response goal**. Service classes can be defined which have a "percentile" response performance goal. A "percentile" response performance goal means that the performance goal is defined as "x%" of the transactions should complete within "y" time. For example, a percentile response goal could be that **90% of the transactions should complete within 200 milliseconds**.

  For service classes which have a percentile response time goal, the Workload Manager determines whether the requisite percent of transactions are ending within the response goal. If the requisite percent of transactions is not ending within the response goal, the system is not meeting performance goals for the service class period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet the performance goal.

- **Execution velocity goal**. Installations may specify an *execution velocity goal* for a service class. An execution velocity goal is a measure of how fast work should run when the work is ready to run, without being delayed waiting for WLM-managed resources. Delays for WLM-managed resources include delays waiting for access to a CPU, delayed waiting for access to processor storage[5], or delayed waiting for server address space creation.

  With OS/390 Release 3, execution velocity can optionally include delays waiting for non-paging DASD I/O operations. Non-paging DASD I/O delays include IOS queue delays, subchannel pending delays, and control unit queue delays. Note that DASD disconnect time is not included in the execution velocity delay calculations.

  With OS/390 Version 2 Release 4, execution velocity can optionally include delays waiting for an initiator (with batch jobs in WLM-managed job classes).

  The purpose of specifying an execution velocity is to allow installations to assign an overall processing importance to the work being processed, when the work has no time-related measure (that is, a response requirement is not associated with the work).

- **Discretionary goal**. A discretionary goal is specified for service classes which do not have a response goal or execution velocity goal[6]. A discretionary goal for a service class simply tells the Workload Manager to process the corresponding workload on a "discretionary" basis. The Workload Manager may allocate resources to discretionary work to optimize resource usage. However, the Workload Manager normally will not allocate resources to discretionary work if allocating the resources would cause any service class to miss its performance goal.

---

[5]Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

[6]Any unit of work not assigned to a service class is assigned to the SYSOTHER default service class. A unit of work might not be assigned to a service class because the work was not identified using the workload classification scheme. The SYSOTHER default service class has a discretionary performance goal.

IBM introduced *discretionary goal management* algorithms with OS/390 Version 2 Release 6. With discretionary goal management, service class periods that are **overachieving** their goals may have their CPU resources “capped” in order to allow some CPU resources to be used by service class periods with discretionary goals. See a more complete discussion of discretionary goal management in Chapter 1.7 of this section.

Chapter 1.4.2: Performance goal importance

Users specify how important it is that the performance goal be met: very important to not important. Five levels of importance can be specified: *Importance 1* (very high) to *Importance 5* (not important). These importance values (1 to 5) may be specified for service classes which have a response goal or an execution velocity goal.

The Workload Manager internally defines **Importance 0** to represent system tasks (the most important work) and **Importance 6** to represent discretionary work (the least important work)[7].

Chapter 1.4.3: Service class periods

A service class may be broken into multiple *performance periods*. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional performance periods (although typically no more than two or three additional performance periods are defined). Each performance period has its own performance goal, goal importance, and duration (except that the duration may not be specified for the last performance period).

An address space (TSO transaction, batch job step, etc.) begins execution in Performance Period 1. The address space will transition from Performance Period 1 to Performance Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction.

Except for enclaves, the "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources. The actual resources used by the address space are adjusted by the CPU, SRB, IOC, and MSO service coefficients defined in the Workload Manager ISPF application panel, to yield the CPU, I/O, and memory "service" requirements of the address space.

[7]The goal importance for discretionary work actually is recorded as **Importance 0** in SMF Type 72 records because Workload Manager developers felt that there was no value in reporting Importance 6 externally. From an external view, the lowest importance (Importance 6) is implicit in the fact that the work is discretionary.

©Copyright 1994, Computer Management Sciences, Inc.

When the address space accumulates more than a specified amount of service, the SRM will reassign the address space to the next lower performance period (if multiple periods are defined)[8]. The amount of service controlling when an address space is reassigned to the next lower performance period is specified by the value of the **Duration** specification in the ISPF definition of the service class performance period.

The normal purpose of defining multiple performance periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources.

Multiple performance periods may not be used with a service class representing subsystem transactions (i.e., a service class defined for transactions executing under CICS/ESA Version 4.1 or under IMS/ESA Version 5). This is because system service units are not accumulated by the SRM for the transactions; the service units are accumulated at the **address space level** (i.e., the CICS region or IMS message processing region). Since service is not collected at the transaction level, the SRM cannot migrate service classes to lower periods for service classes representing subsystem transactions.

Chapter 1.5: Report Classes

Work units may be assigned to a report class via the Workload Classification ISPF panels.

Report classes are similar to report performance groups in earlier versions of MVS, but there are some differences.

- The report classes are similar to report performance groups in that (1) the SRM simply accumulates performance-related and resource-related statistics about the workload, (2) the report classes are not used to control workload, and (3) more than one workload can be assigned to a report class.

- The report classes are different from report performance groups in that a transaction or job can be assigned to only one report class. With versions of MVS prior to MVS/ESA SP5, a transaction or job could be assigned to as many as four[9] report

[8]Note that the address space is not reassigned immediately upon accumulation of the specified amount of service. The reassignment is performed only when the SRM evaluates the address space for changed conditions. Therefore, the SRM cannot determine when the **exact** amount of service specified in the DUR parameter is used; the SRM can only determine when the DUR value is exceeded.

The SRM checks for transactions exceeding their DUR value only when the SRM gains control. Each address space normally is evaluated every three SRM seconds, and reassignment is performed only if the address space has accumulated more than the value specified for the DUR keyword. Consequently, an address space will always accumulate **more** service than specified by the DUR keyword before being reassigned.

[9]A transaction or job can be assigned to as many as five report performance groups beginning with MVS/ESA SP5 (Compatibility Mode).

©Copyright 1994, Computer Management Sciences, Inc.

performance groups, depending on the workload classification scheme contained in IEAICSxx member of SYS1.PARMLIB.

Additionally, while report classes may be defined to report on subsystem transactions (e.g., CICS transactions), the report class information will not contain **resource** data for the subsystem transactions. This is because the subsystem (e.g., CICS) does not report transaction resource use to the Workload Manager and thus no resource use information is available for the report class.

Prior to z/OS V1R2, report class information did not include response time distributions and did not include work manager (e.g., CICS or IMS) delay states. This response and delay state information is provided with z/OS V1R2.

Beginning with z/OS V1R2, the overall structure of report classes has been significantly changed. With z/OS V1R2, report classes (1) can have as many **periods** as are reflected in the service class(es) to which work units associated with report classes are assigned, (2) can be used to report on a subset of transactions running in a **single** service, and (3) can combine work units running in **multiple** service classes periods within one report class period.

- **Report classes can have multiple periods**. Prior to z/OS V1R2, a report class had a single period. With z/OS V1R2, report classes can have as many periods as appear in the associated service class(es) to which work units related to the report class are assigned. For example, if a work unit is assigned to a service class and to a report class, and if the service class has 3 periods, then the report class can have three periods.

- **Report class period related to single service class**. A report class period that is related to a subset of work in a **single** service class, is referred to as a **homogeneous** report class period. This term means that the report class period contains resource and performance information about work units in a **single** service class period, and that the performance goal and goal importance related to the report class period are obtained from a **single** service class period. As described below, with z/OS V1R2, it is possible to identify homogeneous report class periods.

- **Report class period related to multiple service classes periods**. A report class can be associated with work units that are assigned to multiple service class periods. That is, the workload classification scheme can associate work units to **multiple** service classes, while associating these work units to a **single** report class. When a report class is associated to work units in multiple service classes, the report class is referred to as a **heterogeneous** report class. This term means that the report class period contains resource and performance information about work units in **multiple** service class periods.

Heterogeneous report classes can cause incorrect or misleading performance data, since the data collected is based on different goals, importance, or duration as specified for the multiple service class periods. As described below, with z/OS V1R2, it is possible to identify heterogeneous report class periods.

The heterogeneous report class will have a number of periods corresponding to the largest number of periods of any associated service class. For example, if Service Class 1 has two periods and Service Class 2 has three periods, and if work units assigned to Service Class 1 and Service Class 2 were assigned to Report Class 1, then Report Class 1 would have three periods (based on Service Class 2 having three periods).

With z/OS V1R2, SMF Type 72 (field R723CRS1: Report class period flags) contains information that describes whether a report class period is homogeneous or heterogeneous, on an RMF recording interval basis. A report class period is described as heterogeneous if more than one service class period was found contributing to the report class period in an RMF recording interval.

This new information means that it is possible to detect whether a report class period reverts between homogeneous and heterogeneous between RMF recording intervals, depending on whether work units actually executed in the corresponding service class periods during the RMF interval.

Additionally, with z/OS V1R2, SMF Type 72 variable R723CLSC contains the name of the service class that last contributed information to the report class period. The performance goal and goal importance described in SMF Type 72 records for the report class period are obtained from the **last** service class period that contributed to the report class period. This new field can be used to associate the performance information to a specific service class period, which could be particularly useful if a report class period reverted between homogeneous and heterogeneous states from one RMF recording interval to the next.

Chapter 1.6: Resource Groups

A resource group is simply a "named" description of the minimum and maximum unweighted CPU service units per second which may be used by one or more service classes. A resource group applies across an entire sysplex.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

One or more service classes may be associated with a resource group[10]. The Workload Manager will attempt to provide the specified minimum CPU service to the resource group and will restrict the resource group from using more than the specified maximum CPU service. The minimum and maximum specifications apply on a sysplex-wide basis. For example, three resource groups could be defined as:

GROUP NAME	MINIMUM CPU	MAXIMUM CPU
LIMITCPU	0	500
TOP_PRI	100000	0
4381-91E	309	309

- One or more service classes could be associated with the LIMITCPU Resource Group, and the service classes normally would be limited to using a collective **maximum** of 500 CPU service units per second. The limit on access to CPU service is termed "resource capping" in IBM documents.

- In contrast, the TOP_PRI Resource Group specifies a **minimum** CPU service units, and the Workload Manager will attempt to provide the minimum CPU service specified. Service classes could be associated with the Resource Group TOP_PRI and the Workload Manager would attempt to provide a total of 100,000 CPU units per second to those service classes. The Workload Manager would assign a high dispatching priority to service classes associated with the TOP_PRI Resource Group in an attempt to guarantee that they had access to the CPU.

  Practically, of course, it is not possible with current systems to provide 100,000 CPU service units per second. The specification simply means that the Workload Manager would try to provide service classes associated with the TOP_PRI Resource Group with access to the CPU whenever the service classes wanted access. The system on which the service classes were executing might actually be capable of providing a fraction of the specified minimum CPU service.

- The 4381-91E Resource Group defines both a **minimum** and a **maximum** CPU capacity. In this example (taken from IBM's *MVS/ESA SP5.1 Planning: Workload Management* document), the minimum and maximum specifications are the CPU

[10]A service class representing subsystem transactions (i.e., a service class defined for transactions executing under CICS/ESA Version 4.1 or under IMS/ESA Version 5) should not be associated with a resource group. This is because CPU resources are not monitored by the SRM for the transactions; the CPU resources are monitored at the **address space level** (i.e., the CICS region or IMS message processing region). Further, CPU dispatching occurs at the address space level, rather than at the transaction level. Since CPU usage is not collected at the transaction level and CPU dispatching is at the address space level, the Workload Manager cannot control the amount of CPU resources allocated to service classes which represent transactions. Thus, while the Workload Manager ISPF application does not prevent assignment of subsystem transactions to a service class assigned to a resource group, there is no effect in doing so; the Workload Manager ignores the resource group values with regard to the subsystem transaction service class.

©Copyright 1994, Computer Management Sciences, Inc.

service units provided by an IBM-4381-91E processor. With this specification, the Workload Manager will attempt to provide exactly the CPU service which would be provided if the service classes associated with the Resource Group were running on an IBM-4381-91E.

This resource group might be appropriate for a service bureau which has "sold" a 4381-91E equivalent to some customer.

If you should make such a specification to direct the Workload Manager to provide an equivalent amount of CPU time, you probably should not specify the "exact" values listed for the appropriate processor. Keep in mind that work executing on the "equivalent" processor would have experienced MVS overhead, wait on I/O, and other conditions. The resource group minimum and maximum relate to CPU time used by service classes, and do not include MVS overhead or other overhead.

The Workload Manager implements resource group boundaries by a cooperative interaction with the SRM.

- The Workload Manager attempts to provide the minimum CPU service units to the service classes associated with the resource group. The Workload Manager implements the CPU minimum boundary by assigning a high resource allocation priority to the service classes associated with a resource group which is below its specified minimum CPU.

  The high resource allocation may result in a high CPU dispatching priority, a high central storage allocation priority (to minimize page faults), etc.

- The Workload Manager implements the CPU maximum boundary by setting bits in the SRM's control blocks to indicate that address spaces in the service class are "CPU capped" and are not eligible for dispatching[11].

  - The Workload Manager periodically sums the CPU time used by each resource group (that is, the CPU time of all address spaces assigned to the service classes associated with the resource group). The Workload Manager implements CPU capping for the resource group if the CPU time used is greater than the maximum specified for the resource group.

[11]Capslicing is a technique somewhat similar to "timeslicing" used in earlier versions of MVS. Timeslicing is a technique by which each second of real time (adjusted by a processor-dependent constant) is divided into 64 "slices". For the duration of a timeslice, an address space may have its dispatching priority at a relatively low priority or may have its dispatching priority raised to a high priority. This timeslicing facility has been an inherent part of the SRM design. Conceptually, timeslicing as implemented by having a vector of 64 bits associated with the address space, with each bit representing a timeslice. The first bit would represent the first 1/64 slice at the start of a second, while the last bit would represent the last 1/64 slice at the end of a second of real time. If any bit is "ON", the SRM raises the dispatching priority of the address space during the corresponding time slice. The dispatching priority reverts to the original level during the next slice when the corresponding bit is "OFF". Thus, the dispatching priority of an address space can be raised and lowered from a "base" priority to a "timeslice" priority.

©Copyright 1994, Computer Management Sciences, Inc.

- The Workload Manager manipulates a "cap slice" bit vector for the resource group, to prevent all address spaces in all service classes associated with the resource group from using the CPU during the "capped" timeslice interval.

- Each "CPU capped" status represents 1/64 second of real time (adjusted by a processor-dependent constant), and uses the timeslice concept. The difference is that rather than raising the dispatching priority during the time slice as was done in earlier versions of MVS, the SRM marks the address space non-dispatchable during the time slice to restrict the service class use of CPU service.

- It is important to appreciate that **all** address spaces in all service class periods in a resource group are controlled by the **same** number of cap slices. That is, when the resource group is in its "cap slice", all address spaces associated with the resource group are capped. This cap status has no effect if the address space is waiting on some event (waiting on I/O, etc.). However, any dispatchable unit (TCB or SRB) on the dispatch queue will not be dispatched by the MVS Dispatcher if the dispatchable unit has its "cap" bit turned on.

It normally is not advisable to use resource groups. IBM provides the facility solely for special cases, and IBM does not contemplate resource groups normally being used. Resource group specifications are "preemptive" in nature, in that the Workload Manager attempts to honor resource group specifications before considering other service specifications. Consequently, **resource group specifications could nullify the rest of the Workload Manager's algorithms**.

Chapter 1.7: Discretionary Goal Management

A problem existed when using discretionary goals prior to OS/390 Version 2 Release 6: on systems in which 100% of the CPU was used by service class periods with performance goals, service class periods assigned a discretionary goal might never receive CPU service. This situation existed even though the service class periods with performance goals might be significantly **overachieving** their goals, since the Workload Manager would never allow discretionary work to have a CPU dispatching priority equal to or higher than work with performance goals.

From one perspective, this algorithm is proper; discretionary work is defined as work that has no performance goal. However, most sites want the discretionary work eventually to be processed, even though it has no performance goal. Consequently, many sites removed the discretionary goal from work and assigned a performance goal to the work.

However, there are significant advantages to assigning a discretionary goal to work: work with a discretionary goal executes with the Mean-Time-To-Wait (MTTW) algorithm.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

- Work assigned to a Mean-Time-To-Wait group competes within the Mean-Time-To-Wait group for access to the processor. Address spaces are assigned dispatching priority within the MTTW group, based upon their execution characteristics. Address spaces that execute a significant amount of CPU instructions between I/O operations are considered heavy CPU users. These heavy users receive a lower dispatching priority within the MTTW group than do address spaces requiring less CPU processing between I/O operations.

- The philosophy behind assigning work to Mean-Time-To-Wait groups is to attempt to use as much of the overall computer system as possible. Dispatching relatively light CPU users ahead of relatively heavy CPU users ensures that the I/O complex will be used simultaneously with the CPU processor. Since both CPU and I/O are active simultaneously, more overall work will be accomplished by the computer system. This philosophy assumes, of course, that overall throughput is a major goal, rather than the turnaround of specific heavy CPU users. This philosophy is explicitly applicable to service class periods assigned a discretionary goal.

IBM addressed this problem in OS/390 Version 2 Release 6, by implementing the *discretionary goal management* algorithms[12] .

With discretionary goal management, the Workload Manager identifies service class periods that have been assigned a performance goal and which are candidates for participation in discretionary goal management. Service class periods can participate in discretionary goal management if either of the following conditions apply:

- The service class period has a response goal greater than one minute. This condition does not apply to subsystem transaction service classes (e.g., CICS or IMS transaction service classes), since these service class periods do not include address spaces.

- The service class period has an execution velocity goal less than or equal to 30%.

The Workload Manager identifies candidate service class periods meeting either of the above conditions, that have **significantly** overachieved their performance goal. If discretionary work exists in the system, the Workload Manager may apply *internal resource capping* to the service class periods that are overachieving their performance goal. The internal resource capping operates similarly to the normal Resource Group capping described in Chapter 1.6 of this section, in that the Workload Manager will cap the address spaces for one or more cap slices. This capping restricts the amount of CPU service that can be used by address spaces in the capped service class period.

[12]CPExpert is proud to note that Don Deese described the basic discretionary goal management design to IBM SRM/WLM developers (Steve Grabarits and Peter Yukom) at a closed working session of the Enterprise Wide Capacity Management (EWCP) project of SHARE.

The Workload Manager may apply internal resource capping when the Performance Index[13] is less than 0.7, and stops internal resource capping when the Performance Index is greater than or equal to 0.81. If a candidate service class period with a performance goal has multiple periods, later periods are selected for capping before earlier periods (that is, capping would potentially be applied to Period 2 before capping would be considered for Period 1).

The effect of the discretionary goal management algorithm is to allow discretionary work to receive CPU cycles when work with a performance goal would otherwise significantly overachieve its performance goal.

Chapter 1.8: Long-Term Storage Protection

With Compatibility Mode, organizations can specify “storage isolation” for critical work by using the PWSS keyword in the performance group period definition in the IEAIPSxx member of SYS1.PARMLIB. Once storage isolation is specified for a performance group period, the System Resources Manager will not normally steal processor storage from address spaces assigned to the performance group period, unless the processor storage used by the address space is more than the minimum storage amount specified in the PWSS keyword.

This ability to control the potential use of processor storage by work was not available with Goal Mode prior to OS/390 V2R10. Rather, the Workload Manager implicitly controlled the assignment of processor storage, based on how well different service class periods met their performance goal and based on the relative importance of the work. Consequently, some organizations have critical work that can suffer unacceptably under Goal Mode prior to OS/390 V2R10, because there was no ability for the organization to protect storage for the work. The problem typically arises when work goes “dormant” for some period of time.

For example, when there are no transactions being processed by a CICS region for some interval, the CICS region can be considered dormant. Since the CICS region is not processing transactions, the Workload Manger can decide to steal pages from the region. This action might be taken if necessary to provide storage to other work that might require processor storage to meet its performance goal. Unfortunately, when new transactions arrive, the CICS region might have lost much of its processor storage, and considerable paging might be required to restore the stolen pages. Retrieving the pages might considerably delay the new transactions. For critical work, this delay might be unacceptable.

[13]The Performance Index is described in Chapter 3.4 of this section. Note that 0.81 is the Performance Index arbitrarily assigned to work with a discretionary goal.

©Copyright 1994, Computer Management Sciences, Inc.

With OS/390 V2R10, IBM introduced a “long-term storage protection” option. This option is available with APAR OW43810 installed. With the APAR applied, organizations can specify long-term storage protection on the WLM ISPF “Modify Rules for the Subsystem Type” panel. This is done by using the new “Storage Critical” option on this panel.

When long-term storage protection is assigned to 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 will lose storage only to other work of equal or greater importance that needs the storage to meet performance goals.

Storage protection can be assigned to all types of address spaces that are classified for ASCH, JES, OMVS, STC, and TSO. When you specify YES in the new "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 two 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.

CICS and IMS work have another option that is available when you are creating transaction service classes (these are, service classes for which you specify transaction response goals). For transaction service classes, you can **also** assign long-term storage protection by specifying YES in the "Storage Critical" field in the rules for specific transactions.

It is important to appreciate that when 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[14] 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. This is because storage protection is actually implemented at the *address space* level, even though you specify storage protection for a transaction service class. The CICS or IMS *transaction* is not an address space but is processed by the CICS or IMS region, and the region is the address space. Thus, all transactions processed by the region would benefit from storage protection.

As an alternative to assigning storage protection based on specific transaction service classes, you can assign storage protection to the CICS or IMS region which processes the

[14] This would be the case when the region is managed to the response time goals of the transactions it serves.

transactions. You do this by adding or modifying the STC or JES classification rule that assigns the service class to the region.

Please note the potential problem that occurs when you exclude a CICS or IMS region from being managed as a server by the Workload Manager. This option and the potential problem are discussed in Chapter 1.10 below.

Chapter 1.9 Long-Term CPU Protection

With Compatibility Mode, organizations can explicitly control CPU dispatching priority by assigning dispatching to performance group periods, using the DP keyword in the IEAIPSxx member of SYS1.PARMLIB. This ability to control explicitly the CPU dispatching priority by work was not available with Goal Mode prior to OS/390 V2R10. Rather, the Workload Manager explicitly controlled the assignment of CPU dispatching priority, based on how well different service class periods met their performance goal and based on the relative importance of the work[15]. This assignment of dispatching priority based on goal attainment can result in important work having a lower CPU dispatching priority than work that is less important.

For example, suppose that a particular service class is exceeding its service goal and another less-important service class is missing its performance goal. Further suppose that CPU delay is the reason that the less-important service class is missing its performance goal. After some analysis, the Workload Manager might decide to treat the high-importance service class as a “donor” of CPU time. The result might be a readjustment of CPU dispatching priorities, with the less-important service class being given a higher CPU dispatching priority than the important service class.

Perhaps with a dynamically changed workload (for example, a surge of transactions being submitted to the high-importance service class), the CPU demands of the high-importance service class might increase. Unfortunately, it might take the Workload Manager some number of policy adjustment intervals to completely reverse its decisions with respect to relative CPU dispatching priority. Consequently, some time might lapse[16] before the Workload Manager would adjust CPU dispatching priorities such that the high-importance was at a higher dispatching priority than the less-important work.

Consequently, some organizations have critical work that can suffer unacceptably under Goal Mode prior to OS/390 V2R10 because there was no ability for the organization to specify long-term CPU protection for the work.

---

[15] There are situations, unrelated to attainment of performance goals or importance of the work, when the Workload Manager can control the dispatching of work. These situations include the infamous “small CPU users” concept in which the Workload Manager can assign a very high dispatching priority to a dispatchable unit if it believes that the dispatchable unit will consume only a small amount of CPU time before entering a wait state. Other more rational situations include giving a high CPU dispatching priority to work that holds an enqueue. In both cases, this action is taken in an attempt to quickly move the work through the CPU.

[16] Recall that a policy adjustment interval is 10 seconds.

With OS/390 V2R10, IBM introduced a “long-term CPU protection” option. This option is available with APAR OW43855 installed. With the APAR applied, organizations can specify long-term CPU protection on the WLM ISPF “Modify Rules for the Subsystem Type” panel. This is done by using the new “CPU Critical” option on this panel.

When long-term CPU protection is assigned to work, WLM normally ensures that less important work will have a lower CPU dispatching priority than the work with CPU protection. This option can be useful for critical work that needs to retain high CPU dispatching priority relative to other work, because the work is extremely CPU-sensitive. This requirement would exist regardless of whether the critical work meets or exceeds its performance goal (that is, there should be no situation when less-important work would preempt the critical work from access to a CPU)[17].

CPU protection can be assigned to service classes with address spaces or with enclaves. However, the service class must:

- Have only one period, and
- The service class cannot have a discretionary goal.

Additionally, CICS and IMS work have another option that is available when you are creating transaction service classes (these are, service classes for which you specify transaction response goals). For transaction service classes, you can **also** assign long-term CPU protection by specifying YES in the "CPU Critical" field in the rules for specific transactions.

It is important to appreciate that when you specify YES for **one** transaction in a CICS/IMS service class, **all** CICS/IMS transactions in that service class will be CPU-protected.

If a CICS or IMS region is managed as a server by WLM[18] 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. This is because CPU protection is actually implemented at the *address space* level, even though you specify CPU protection for a transaction service class. The CICS or IMS *transaction* is not an address space but is processed by the CICS or IMS region, and the region is the address space. Thus, all transactions processed by the region would benefit from CPU protection.

As an alternative to assigning CPU protection based on specific transaction service classes, you can assign CPU protection to the CICS or IMS region which processes the

[17] Note that there still can exist situations when other work is temporarily given a higher CPU dispatching priority. For example, the “small CPU consumer” algorithm still applies, and work might be given a high dispatching priority because the work is holding an enqueue for which there is contention. Thus, there is no absolute guarantee that critical work will not have less-important work assigned a higher CPU dispatching priority. However, less-important work would not be assigned a higher CPU dispatching priority as a result of the normal Workload Manager policy adjustment algorithms.

[18] This would be the case when the region is managed to the response time goals of the transactions it serves.

transactions. You do this by adding or modifying the STC or JES classification rule that assigns the service class to the region.

Please note the potential problem that occurs when you exclude a CICS or IMS region from being managed as a server by the Workload Manager. This option and the potential problem are discussed in Chapter 1.10 below.

Chapter 1.10: Exemption from server management

If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

The concept of assigning subsystem transactions to service classes is described in detail in Chapter 2 of this section.

One problem with the subsystem transaction concept is the design that once a single subsystem transaction is assigned to a service class, *all transactions entering the system must be classified to either the defined service class or must be classified to a* ***default*** *service class that must be specified.* For example, if a single CICS transaction is assigned to a transaction service class, all CICS transactions must be assigned to either that transaction service class or assigned to a default transaction service. Additionally, the default transaction service class must have a response time goal specified. Further, **all CICS regions processing the transactions** will be managed by the Workload Manager based on how well the transactions (in both the defined transaction service class and the default transaction service class) are meeting their performance goal.

This is often referred to as the “all or nothing” approach: you must either classify all subsystem transactions to transaction service classes with appropriate performance goals and goal importance, or you may not assign any subsystem transactions to transaction service classes.

The “all or nothing” approach has caused many installations running under Goal Mode to refrain from classifying subsystem transactions, since some transactions do not lend themselves to response time goals. These installations have simply specified execution velocity goals for all CICS or IMS regions and not classified any subsystem transactions to transaction service classes. Consequently, one of the significant potential benefits of Goal Mode (the ability to manage CICS or IMS transactions based on response goals) has not been realized by these organizations.

With OS/390 V2R10, IBM introduced an “exemption from transaction response time management” option. This option is available with APAR OW43812 installed. With the

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

APAR applied, organizations can specify whether an address space (CICS region or IMS region) will be managed based on the goals of the transactions that the region is serving, or managed based on the goals specified for the region itself. This option is exercised by using the new "Manage Region Using Goals Of:" field on the WLM ISPF "Modify Rules for the Subsystem Type" panel.

When "TRANSACTION" is entered in the "Manage Region Using Goals OF:" field, the region will be managed as a CICS/IMS transaction server by the WLM. "TRANSACTION" is the default specification. If "REGION" is entered in this field, the region will be managed based on the performance goal specified for the service class to which the region is assigned. This performance goal normally would be an execution velocity goal.

When "REGION" is specified, the WLM does not consider the region to be a "server" of transactions[19]. Rather, the WLM server topology algorithms ignore the region when establishing server topology. Consequently, the goals for any transaction processed by the region will not be considered by the WLM when it determines whether service class periods meet goals and whether policy adjustment is necessary.

This consequence might have undesired implications if you specify goals for CICS or IMS transactions and some or all of those transactions are processed by a CICS or IMS region that has "REGION" specified in the "Manage Region Using Goals Of:" field. In this case, **performance of the transaction service class will not be considered when adjusting resource policy for the region**. This could have the undesired result of transactions not achieving the performance that you desire, simply because the transactions were processed by a CICS or IMS region that was managed based on the goals specified for the region. Alternatively, some transactions might receive better performance than desired because of the same "region-oriented" management by the WLM.

[19] Please refer to Chapter 2 (Subsystem Transactions) for a discussion of the servers and served concept.

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 2: Subsystem Transactions

If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

Chapter 2.1: Subsystem transaction service classes

Exhibit 4-2 shows a sample of several service classes. Included in Exhibit 4-2 are CICSAOR1, CICPERS, CICADMN, and CICOPS. While Exhibit 4-2 does not identify the nature of the service classes, suppose that the CICSAOR1 service class describes a CICS region while the CICPERS, CICADMN, and CICOPS service classes describe CICS transactions.

In the example, an installation would have defined classification rules to the Workload Manager so all CICS transactions from the Personnel department would be assigned to the CICPERS Service Class, all CICS transactions from the Administrative Department would be assigned to the CICADMN Service Class, and all CICS transactions from the Operations Department would be assigned to the CICOPS Service Class.

The installation could define specific performance goals for transactions from the Personnel Department, the Administrative Department, and the Operations Department. The installation could specify goal importance for the different goals.

The controlling address space (e.g., the CICS region) must be in its own service class. In our example, suppose that the CICS region was placed into the CICSAOR1 Service Class. The CICSAOR1 Service Class would be considered a "server" and the CICPERS Service Class, CICADMN Service Class, and CICOPS Service Class would "served" service classes controlled by the CICSAOR1 Service Class.

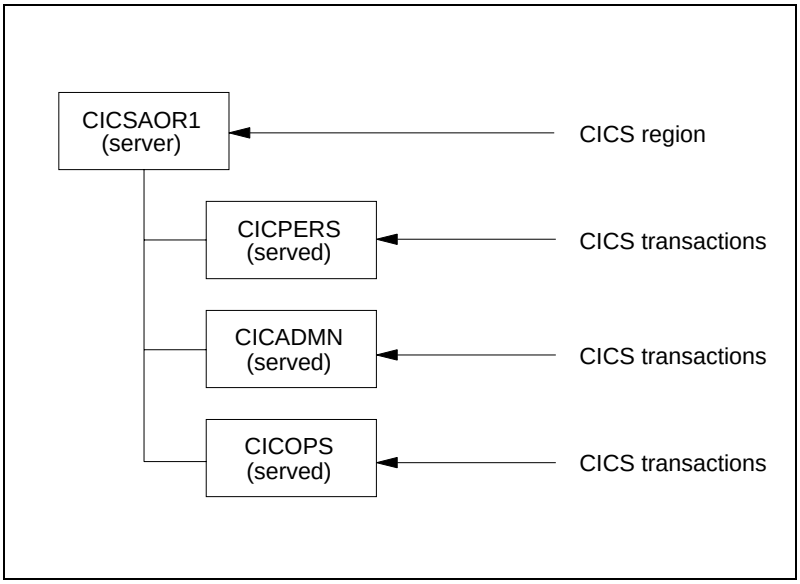
Exhibit 4-3 illustrates the relationship between the service classes

Notice that the **transactions** comprising the CICSPERS, CICSADMN, and CICSOPS Service Classes must actually execute under control of a CICS region (CICSAOR1 in our example) executing CICS at a level of at least CICS/ESA Version 4.1. The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The CICSAOR1 will have its own performance goals and importance. However, these performance goals and importance normally are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service and classes (in our example, the allocation will be based upon the performance of the CICPERS transactions, CICADMN transactions, and CICOPS transactions)[20].



RELATIONSHIP BETWEEN SERVER AND SERVED

Exhibit 4-3

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICPERS/CICADMN/CICOPS Service Classes, as these service classes are simply logical entities which describe transactions and are not address spaces. Rather, the Workload Manager allocates resources to the "server" address space (the CICSAOR1 Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICPERS/CICADMN/CICOPS Service Classes, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" service classes which are served by the "server" service class. It does this by allocating resources to the "server" service class. **These additional resources may (or may not) be used by CICS to provide service to the service class missing its goal**.

[20]This statement is not **strictly** true. If the CICS region should become idle for an extended period (no transactions executed in the "served" service classes), the Workload Manager would use the service goal and importance specified for the CICS region service class to manage the region. Practically, of course, there would be little to manage with an idle region.

©Copyright 1994, Computer Management Sciences, Inc.

Suppose there are multiple "served" service classes associated with a "server" service class (CICPERS, CICADMN, and CICOPS in our example). If some "served" service class is failing to achieve its goal, the Workload Manager may allocate additional resources to the "server" service class. These additional resources might allow some "served" service classes to significantly exceed their performance goal and these "served" service classes may not be particularly important.

In our example, the CICSAOR1 Service Class is serving three service classes (CICPERS, CICADMN, and CICOPS).

- Suppose that CICPERS is important but that CICADMN Service Class is of lower importance.
- If the Workload Manager detects that CICPERS is not meeting its performance goal, the Workload Manager may allocate more resources to the CICSAOR1 Service Class.
- The CICSAOR1 would use the additional resources to provide service to CICPERS, to CICADMN, and to CICOPS.
- Consequently, CICADMN or CICOPS might significantly exceed their performance goals[21], depending upon how the CICSAOR1 region provided internal dispatching.

To summarize this discussion, performance goals are associated with "served" service classes while resources are allocated to "server" service classes. Performance (i.e., transaction response time) is recorded at the "served" service class level, while resource use is recorded at the "server" service class level.

Chapter 2.2: Subsystem interaction with Workload Manager

Subsystems (e.g., CICS or IMS) communicate with the Workload Manager using Workload Manager Services macros. These macros are described in IBM's *Programming: Workload Management Services* document. This chapter presents a brief overview of the interaction between the subsystem and the Workload Manager.

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[22].

[21]Indeed, there is no guarantee that the additional resources would help CICPERS unless the tasks supporting the transactions assigned to CICPERS had been properly **defined to CICS** as a higher priority than CICADMN. Further, tasks might be common between CICPERS and CICADMN (for example). In this case, there is no way to help CICPERS even within CICSAOR1.

[22]IMS Version 5 reports only *execution phase* samples.

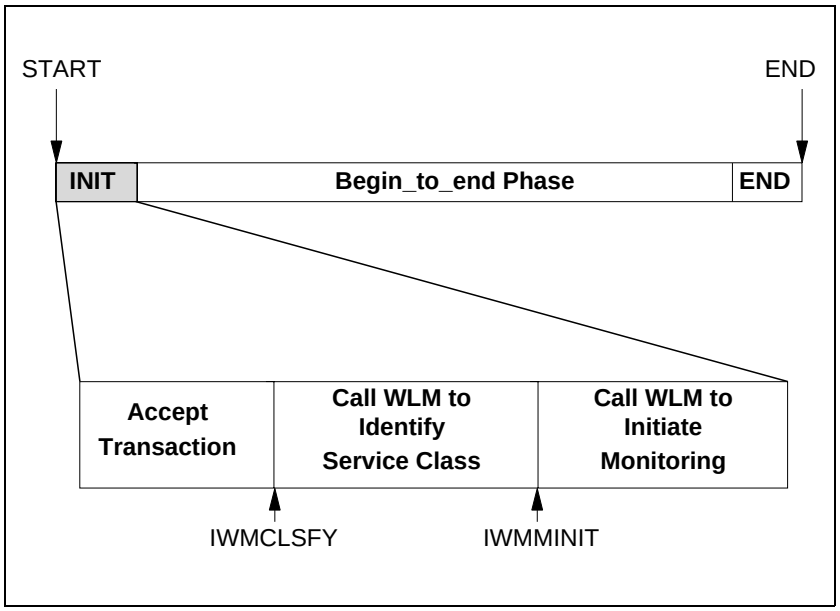
- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

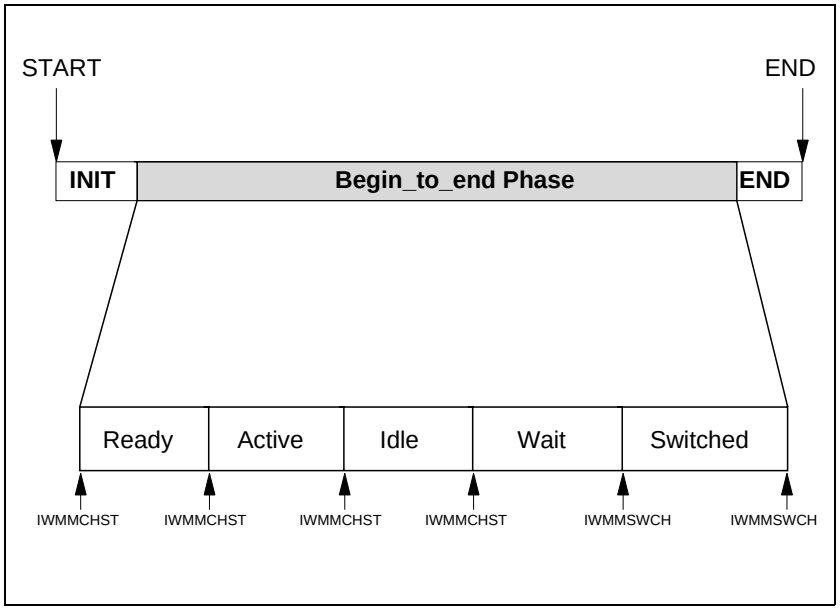
The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase. Exhibit 4-4 illustrates the sequence of events.



TRANSACTION INITIALIZATION

Exhibit 4-4

CICS or IMS will provide the SRM with information about the "state" of the transaction by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of the transaction. Exhibit 4-5 illustrates the sequence of events.



SUBSYSTEM TRANSACTION STATES

Exhibit 4-5

The CICS/ESA subsystem work manager reports transaction delays in the following states for the "served" service class:

- **Ready state**. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to process transactions in the "served" service class than were dispatched by CICS.

- **Active state**. The active state indicates that there was a program executing on behalf of the work request in the "served" service class, from the perspective of the work manager. In the case of a CICS region, this means that a CICS task has been dispatched by CICS to process the transaction.

  However, the active state **does not mean that the task is executing** from the perspective of MVS. It simply means that the task has been dispatched by CICS. Other address spaces with a higher system dispatching priority could preempt the task dispatched by CICS, and these other address spaces could be using the CPU. The

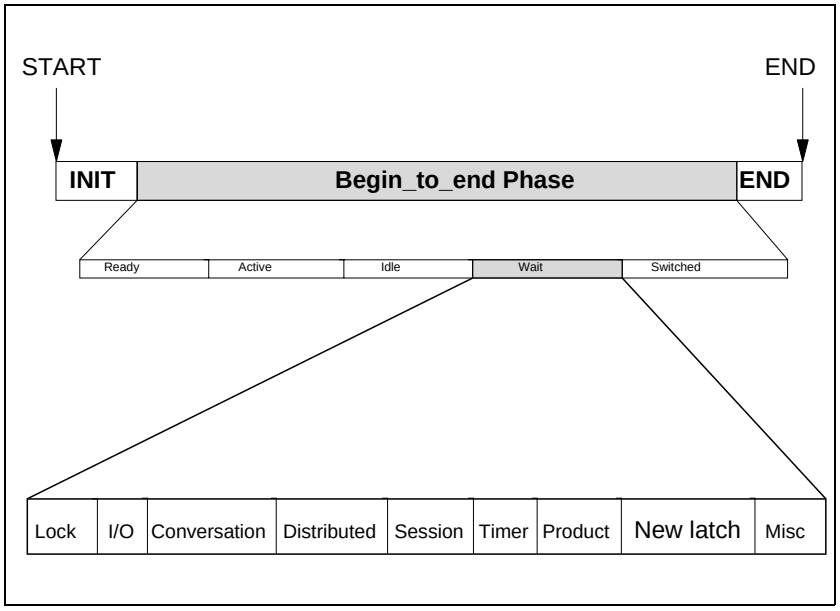
situation in which the CICS application task is denied use of the CPU is unknown to CICS.

- **Idle state**. The idle state indicates that there were no work requests (e.g., CICS transactions) ready to run in the service class.

- **Wait state**. The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the sysplex, or somewhere in the network), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

- **Switched state**. The Switched state indicates that processing of the transaction had been switched from the CICS or IMS work manager providing information to the Workload Manager. The transaction could have been switched to another CICS region (for example) in the same MVS image, switched to another MVS image in the sysplex, or switched to somewhere in the network.

The Wait State is broken into several categories. Exhibit 4-6 illustrates these categories.



WAIT STATE CATEGORIES

EXHIBIT 4-6

- **Waiting for lock**. The waiting for lock state indicates that some work request (e.g., a CICS task) was waiting for a lock.

- **Waiting for I/O**. The waiting for I/O state indicates that the work manager was waiting for some I/O request on behalf of the "served" service class. This state could be waiting on an actual I/O operation or waiting on some other function related to the I/O request.

- **Waiting for conversation**. The waiting for conversation state indicates that the work manager was waiting for a response in a conversation mode.

- **Waiting for distributed request**. The waiting for distributed request state indicates that some function or data must be routed prior to resumption of the work request.

- **Waiting for session to be established locally**. The waiting for session to be established locally means a wait for a session to be established on the current MVS image.

- **Waiting for session to be established in sysplex**. The waiting for session to be established in sysplex means a wait for a session to be established somewhere in the sysplex.

- **Waiting for session to be established in network**. The waiting for session to be established in network means a wait for a session to be established somewhere in the network.

- **Waiting for timer**. The waiting for timer means that a work request was waiting for expiration of a timer.

- **Waiting for another product**. The waiting for another product means that a work request was waiting for another product to provide some service.

- **Waiting for a new latch**. The waiting for a new latch means that a work request was waiting for a new latch. A latch is a short-duration lock.

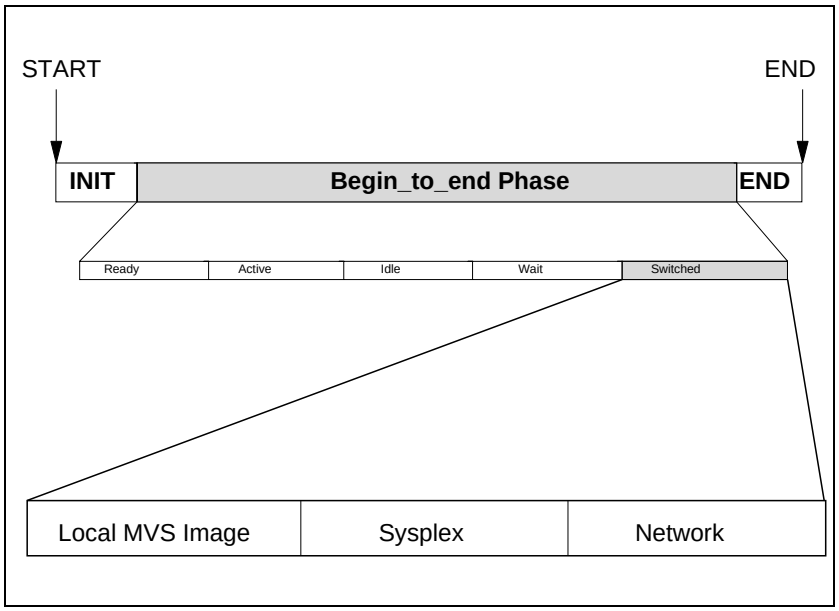
- **Waiting for unidentified resource**. The waiting for unidentified resource means that the work request was waiting, but that the work manager could not identify the cause of the wait.

Additionally, the Switched State is broken into three categories, describing where the transaction has been switched. Exhibit 4-7 illustrates these categories.

Many CICS transactions are function-shipped from a CICS Terminal Owning Region (TOR) to a CICS Application Owning Region (AOR). The time spent in the TOR is considered

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

the *begin_to_end phase*. The *execution phase* begins once the transaction is shipped to an AOR.



SWITCHED STATE CATEGORIES

Exhibit 4-7

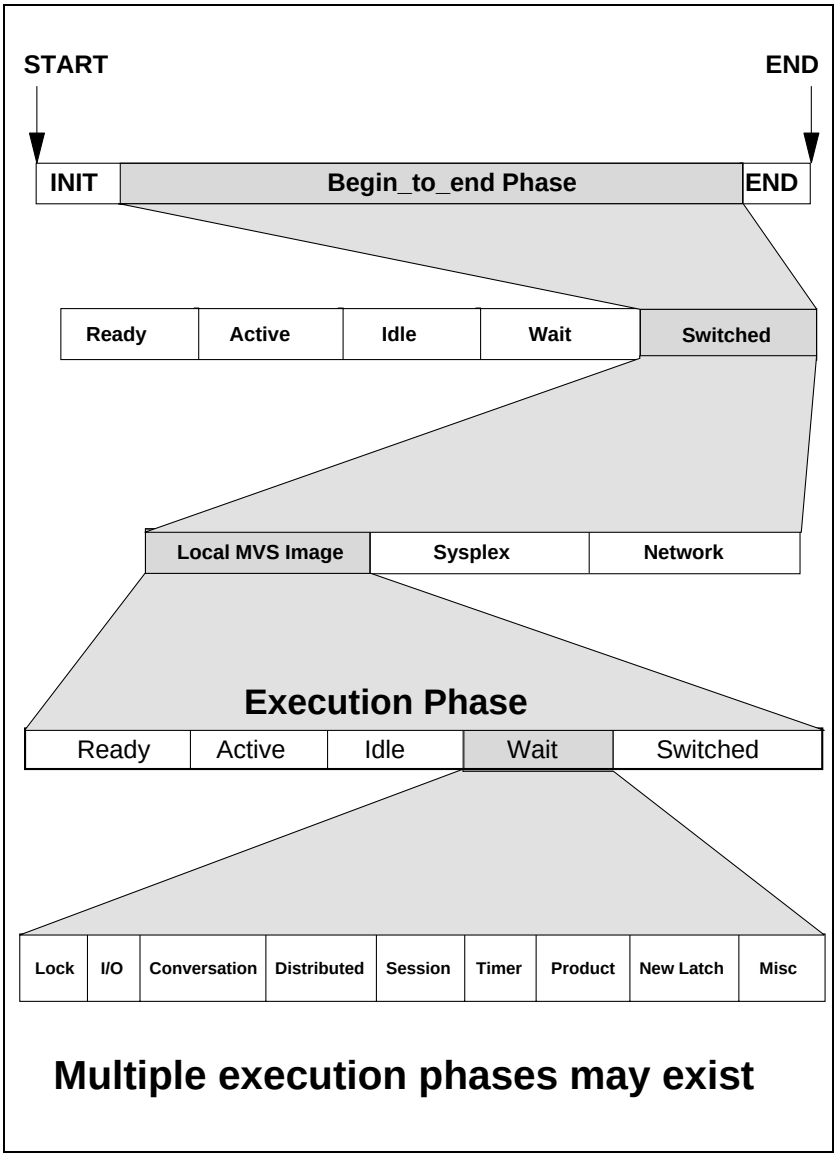
When the transaction is switched to another subsystem (switched from a CICS TOR to a CICS AOR, for example) in the same MVS image or switched to another MVS image in the sysplex, the subsystem from which the transaction is being shipped indicates that the monitoring environment transaction is being transferred to another subsystem. In this situation, the receiving subsystem provides transaction delay information to the Workload Manager.

Exhibit 4-8 illustrates the execution phase information for a local MVS image. The same information would be available if the receiving subsystem was on another system in the sysplex.

If the transaction is switched somewhere in the network. the Workload Manager has no more information about the status of the transaction; it is simply "switched in the network" from the Workload Manager's view.

Notice that the transaction state information provided in the execution phase is the same as the information provided in the begin_to_end phase. However, the Workload Manager is notified about the phase of the transaction. The transaction state information is maintained separately (and reported separately) for the begin_to_end phase and the execution phase.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001



EXECUTION PHASE STATES

Exhibit 4-8

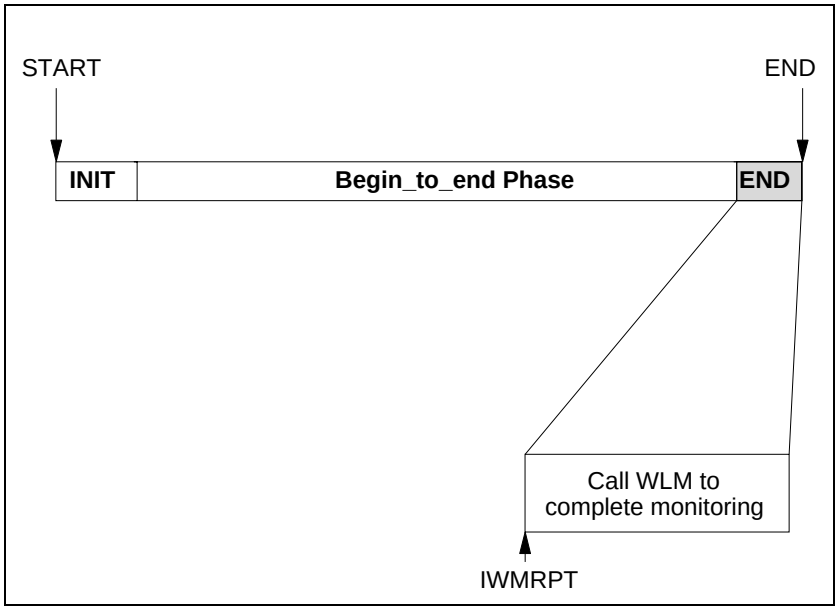
The level of detail about the state of a transaction in the execution phase depends upon where the transaction is switched.

- If the transaction is switched in the local MVS image to another CICS region, the transaction state information for the various CICS regions is combined in the "CICS subsystem" information.

- If the transaction is shipped to an IMS region (perhaps for data access), the IMS subsystem information is provided separately under the "IMS subsystem" information.

- If the transaction is switched to another MVS image in the sysplex, the information would be separately identified by the Workload Manager executing on that MVS image.

Once a transaction has completed the execution phase, the transaction eventually is switched back to the subsystem which received and classified the transaction (typically a CICS TOR). The begin_to_end phase of the transaction ends when this subsystem notifies the Workload Manager to end monitoring of the transaction. Exhibit 4-9 illustrates this sequence.



END OF BEGIN_TO_END PHASE

Exhibit 4-9

The above discussion illustrates three significant times may be associated with a CICS transaction:

- The total elapsed time of the transaction (from START to END)
- The begin_to_end phase (time spent in a TOR)
- The execution phase (time spent in an AOR).

Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. These CICS transactions are termed "non-routed" transactions. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 3: Workload Manager Internal Logic

The Workload Manager operates on a cooperative basis with the System Resources Manager (SRM). The SRM provides information for the Workload Manager's analysis, the Workload Manager alters SRM control blocks based on its analysis of the information, the Workload Manager issues SYSEVENTs to invoke SRM logic, and the SRM manages system resources based on the contents of the control blocks.

The SRM, in turn, operates on a cooperative basis with other parts of MVS, such as the Real Storage Manager (RSM) and the Auxiliary Storage Manager (ASM) to manage address spaces and manage access by the address spaces to system resources.

This chapter briefly describes the internal logic of the Workload Manager and its interaction with the System Resources Manager.

Chapter 3.1: Resource Use Information

The SRM maintains control blocks to describe each address space under its control.

- The control blocks contain information describing the state of the address space (whether it is ready, waiting, logically swapped, swapped to expanded storage, swapped to auxiliary storage, etc.). The SRM updates the state of each address space as it manages the address spaces (for example, when an address space is swapped, the SRM updates appropriate control block variables).

- The control blocks contain information describing the resources held or used by the address space (the amount of CPU resources used, the number of I/O operations executed, the number of central storage frames held, the number of expanded storage frames held, etc.). Resource use information is placed into SRM control blocks by other parts of MVS. For example, the MVS Dispatcher updates CPU usage information for each dispatchable unit (i.e., TCB and SRB) associated with an address space, the MVS Real Storage Manager updates central and expanded storage information for each address space, and the MVS Auxiliary Storage Manager updates swap/page information for each address space.

- The control blocks contain information describing the page frame referencing characteristics of each address space (the highest Unreferenced Interval Count of any page in central storage, the highest Unreferenced Interval Count of any page in expanded storage, the number of pages at different values of Unreferenced Interval Count). The Real Storage Manager updates the page frame characteristics of each address space.

- The control blocks contain information used by the SRM to control each address space's access to system resources (CPU dispatching priority, the domain to which the

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

address space is assigned, the CPU resource cap timeslicing pattern, whether page stealing can be used by or against the address space, etc.). The SRM may update the control information or the Workload Manager may update the control information.

Chapter 3.2: Domain Control Information

The SRM maintains system-related control blocks to describe domains to which address spaces are assigned (the minimum and maximum multiprogramming levels associated with the domain, the In-target for the domain and Out-target for the domain, the number of users in the domain and their status, workload manager swap recommendation value, etc.).

When a service policy is activated, the Workload Manager causes the SRM to generate a domain for each service class period in the service policy. The Workload Manager generates domain specifications based on the defaults for the performance goals of each service class.

Chapter 3.3: Service Class Period Information

The Workload Manager generates control blocks to describe each service class period defined in the service policy. These control blocks describe the performance goal, the importance of the goal, the resource group (if any) to which the service class is assigned, etc.

Additionally, the Workload Manager generates control blocks for each service class period to describe the resources used by the service class period, delays to the service class period, etc.

The Workload Manager periodically examines the SRM control blocks describing each address space and acquires samples[23] describing the state of each dispatchable unit of an address space (that is, each TCB or SRB associated with the address space). The Workload Manager accumulates the samples into counters which describe the state of the address space[24]. The samples are summarized by service class period.

- **CPU Using**. This state means that the address space was using the CPU.

---

[23]With MVS/ESA SP5.1 Goal Mode, the sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

[24]Address spaces can have more than one dispatchable unit (that is, more than one TCB or SRB). If an address space did have more than one dispatchable unit, the Workload Manager would accumulate state samples to describe the state of each dispatchable unit.

To simplify the discussion, we shall refer to the sample states as being the "state" of the address space, with readers understanding that multiple dispatchable units could simultaneously be waiting on multiple events, or could be using the processor. Thus, for any single sampling, an address space could be counted in more than one state.

- **CPU delay**. This state means that the address space was ready to use the CPU, but was denied access because of dispatching priorities.

- **CPU Capping delay**. This state means that the address space/enclave wanted to use the CPU but it could not because it was capped at that moment. The maximum CPU service units had been consumed for the Resource Group to which the service class was assigned, and the Workload Manager had marked all address spaces or enclaves associated with the Resource Group as non-dispatchable for some cap-slice intervals. This delay does not necessarily mean that address spaces or enclaves in a capped service class had consumed the CPU service units. The CPU service units could have been used by another service class if more than one service class had been assigned to the Resource Group.

  The CPU Capping delay state applies only to dispatchable units (TCBs and SRBs) actually on the dispatcher queue awaiting dispatch. The CPU Capping delay state does not apply to dispatchable units waiting for some other event (waiting for I/O, waiting for ENQ, etc.).

  IBM introduced *discretionary goal management* algorithms with OS/390 Version 2 Release 6. With discretionary goal management, service class periods that are **overachieving** their goals may have their CPU resources "capped" in order to allow some CPU resources to be used by service class periods with discretionary goals. See a more complete discussion of discretionary goal management in Chapter 1.7 of this section.

- **Swap-in delay**. This state means that the address space was delayed on swap-in (the swap-in had started, but had not completed). These delays should be non-existent or extremely infrequent, unless serious problems exist with the auxiliary storage subsystem.

- **MPL delay**. This state means that an address space was ready to be swapped in, but that the SRM had not initiated a swap-in because of target MPL constraints associated with the domain. Recall that there is a domain created for each service class period. Thus, the Workload Manager had imposed MPL constraints on the service class period to which the address space belonged.

- **Private area page-in from auxiliary storage delay**. This state means that the address space was experiencing page faults in the private area and the pages were coming from auxiliary storage.

- **Common area page-in from auxiliary storage delay**. This state means that the address space was experiencing page faults in the Common Area and the pages were coming from auxiliary storage.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

- **Cross-memory page-in from auxiliary storage delay**. This state means that the address space was experiencing page faults in cross-memory access and the pages were coming from auxiliary storage.

- **VIO page-in from auxiliary storage delay**. This state means that the address space was experiencing page faults in VIO and the pages were coming from auxiliary storage.

- **Standard hiperspace page-in from auxiliary storage delay**. This state means that the address space was experiencing page faults in standard hiperspace and the pages were coming from auxiliary storage.

- **ESO hiperspace page-in from auxiliary storage delay**. IBM has defined this state to mean that the address space was experiencing page faults in ESO hiperspace and the pages were coming from auxiliary storage. Pages in ESO hiperspace are, by definition, resident only in expanded storage (ESO = Expanded Storage Only), and are never migrated to auxiliary storage. IBM offers the following explanation[25]:

  "The execution delay for ESO hiperspaces is a calculated value based on the assumption that if an application does a read for an ESO hiperspace page and that page is no longer available (has been cast out), the application will read the data from DASD somewhere.

  WLM/SRM takes the number of times a read failed in this way and multiplies it by the number of delay samples we expect a read of a page from DASD to represent and report the product as the execution delay samples for ESO hiperspace. This obviously is not a perfect solution, but we needed some way to get an estimate of how much delay is caused to an address space by not having enough expanded for an ESO hiperspace. Such an estimate is needed to properly manage the amount of expanded owned by the address space to the address space's goal."

- **Unknown delay**. This state means that the Workload Manager was unable to identify the cause of delay. In practice, this means that the delay was caused by something over which the SRM had no control (e.g., I/O operations not under control of the SRM, ENQ delay, etc.).

- **Idle**. This state means that the address space was idle (the address space was in STIMER wait, TSO terminal wait, or APPC wait; or that an initiator was idle).

Several new sampled values were introduced with OS/390 Version 1 Release 3:

[25]IBM TALKLink RMF FORUM appended at 15:39:18 on 95/05/29 GMT (by YOCOM at KGNVMC)
Subject: Workload Activity Report. Reproduced with permission of the author.

- **I/O using**. This state means that work was found to be using non-paging DASD I/O resources. Within this context, the work is using the resources in either a device *connect* state (transferring data) or in a device *disconnect* state (seeking and latency).

- **I/O delay**. This state means that the work was delayed for non-paging DASD I/O. The delays include IOS queue, subchannel pending, and control unit queue delays.

Additional sampled values were introduced with OS/390 Version 2 Release 4:

- **Total delay samples**. The total delay samples is the sum of all WLM-managed delays. The total delay samples value includes batch queue delay (regardless of whether the service class period contained batch jobs that were run in WLM-managed initiators). .

- **Total execution samples**. The total execution samples is the sum of the total Using samples (CPU Using samples and I/O Using samples), total Delay samples, Unknown samples, and Idle samples. The total execution samples value includes I/O using samples and I/O delay samples (regardless of whether I/O using and delays were included in the execution velocity calculations).

Please note that the Workload Manager cannot acquire address space state samples for all service classes it is attempting to manage. With CICS/ESA Version 4.1 and IMS/ESA Version 5, a service class may be defined to describe CICS or IMS transactions. The workload classification schemes can identify and assign transactions to particular service classes. Users can define performance goals and importance for these transaction service classes.

However, the transactions are not represented by the SRM as an address space. Rather, the transactions are managed by their subsystem (CICS or IMS). The **subsystem** is an address space (CICS region or IMS message processing region). The SRM maintains information (resource use and address space delay) about the subsystem address space, but does not maintain information about the transactions.

CICS/ESA Version 4.1 and IMS/ESA Version 5 subsystems communicate transaction response information to the Workload Manger (using the Workload Manager Services macros). Consequently, the Workload Manager can determine whether the transactions are meeting their performance goals.

However, the Workload Manager cannot directly manage resources to the transaction service class to help transactions achieve performance goals. Rather, the Workload Manager can only change SRM control blocks related to the address space controlling the transactions (that is, the Workload Manager changes SRM control blocks related to the CICS or IMS region).

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

Chapter 3.4: Performance Index

The Workload Manager acquires and analyzes information which allows it to assess whether service classes are meeting their performance goal.

- For response goals, the Workload Manager collects and analyzes **transaction response** information.

- For execution velocity goals, the Workload Manager collects and analyzes **CPU using** information and **processor storage delay** information. With OS/390 Release 3, the Workload Manager collects and *optionally* analyzes **I/O using** and **I/O delay** information.

The Workload Manager periodically assesses the performance of each service class, comparing the performance **achieved** by the service class against the performance goals **specified** for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource allocation policies based on whether service classes are meeting performance goals.

The comparison of performance achieved is accomplished by computing a *Performance Index* for each service class.

- For *average response* goals, the Performance Index is computed by dividing the **actual** response, by the response **goal**. If actual is less than the goal, the Performance Index will be less than one. If actual is greater than the goal, the Performance Index will be greater than one.

  For example, suppose that a response goal of 100 milliseconds had been specified. Further suppose that the actual response was 50 milliseconds. Dividing the actual by the goal would yield a Performance Index of 0.5 (50/100=0.5). However, suppose that the actual response was 250 milliseconds. Dividing the actual by the goal would yield a Performance Index of 2.5 (250/100=2.5).

- For *percentile response* goals, the Performance Index also is computed by dividing the **actual** response by the response **goal**. However, since the goal is a percentage of transactions meeting the response goal, a more complicated algorithm must be used to compute the Performance Index:

  - The Workload Manager calculates the number of transactions required to meet the percentile goal, by multiplying the percentile goal times the number of ending transactions. For example, suppose that the response goal had been stated as "90% of the transactions completing in less than 100 milliseconds". If 200 transactions ended, 180 transactions (0.90 * 200 = 180) must end in less than 100 milliseconds to meet the goal.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

- Using the response time distribution counters maintained by the Workload Manager, the Workload Manager determines the response time which was **achieved** by the number of transactions required to **meet** the percentile goal. This response time is the "actual" response used in calculating the Performance Index.

  Using the above example, suppose that a distribution of transaction response showed that 180 transactions achieved a response of 150 milliseconds or less. 150 milliseconds would be the "actual" response used in the calculation of the Performance Index. The Performance Index would be computed as actual/goal, or 150/100 = 1.5 (this particular example results in performance worse than the goal, since 90% of the transactions failed to achieve 100 milliseconds or less).

- For *execution velocity* goals, the Performance Index is computed by dividing the **goal** by the **achieved** velocity. If actual is greater than the goal, the Performance Index will be less than one. If actual is less than the goal, the Performance Index will be greater than one.

  For example, suppose that an execution goal of 30% had been specified. Further suppose that the actual execution velocity achieved was 50%. Dividing the goal by the actual would yield a Performance Index of 0.6 (30%/50%=0.6), indicating that the service class had met its goal. However, suppose that the actual execution velocity achieved was only 15%. Dividing the goal by the actual would yield a Performance Index of 2.0 (30%/15%=2.0), indicating that the service class had not met its goal.

- For *discretionary* goals, the Performance Index is arbitrarily established as 0.81 by the Workload Manager. The 0.81 Performance Index value means that service class periods with a discretionary goal are always considered to be meeting their goal from the view of the Workload Manager.

Exhibit 4-10 summarizes the results from the above discussion. A Performance Index less than one implies that a performance goal **has** been met, while a Performance Index greater than one implies that a goal has **not** been met. A Performance Index of exactly one implies that the performance goal has been **exactly** met.

PI > 1 => performance worse than goal
PI = 1 => exactly on goal
PI < 1 => performance better than goal

PERFORMANCE INDEX EFFECT

Exhibit 4-10

The Performance Index can be used to compare the performance of service classes, regardless of the type of performance goal specified for the service class[26]. This approach has the appeal of creating a **single number** which can be used to evaluate the performance of service classes based upon how well or how poorly the service classes meet their performance goals.

Chapter 3.5: Policy Adjustment

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource allocation policies based on whether service classes are meeting performance goals. The policy adjustment is performed every 10 seconds.

As the initial step of policy adjustment, the Workload Manager implements the specifications of any resource groups (minimum and maximum CPU service units). **This step is performed regardless of the importance of the service class periods associated with the resource group**.

The Workload Manager initially assesses performance based on the *sysplex performance index* computed for each service class period. This assessment is done at each goal importance level. Policy adjustment actions are evaluated for the worst-performing service class period at the highest goal importance, then the next worst-performing, etc. It is important to realize that only one service class period will be "helped" by the policy adjustment algorithms per policy adjustment interval[27].

If the Workload Manager has evaluated the performance of all service class periods at the highest goal importance *based on sysplex performance index* and no action has been taken, the next step depends on whether APAR OW25542 has been applied.

- **OW25542 has not been applied**. With the normal logic, the Workload Manager will examine the performance of all service class periods at the next-highest goal importance *based on sysplex performance index*. The Workload Manager will continue analyzing performance at successively lower goal importance levels, based on sysplex performance index. After the performance of all service class periods with goals have been analyzed with no action, the Workload Manager will perform the analysis beginning with the highest goal importance, **using the local performance index** as the measure of performance,.

[26]A discretionary goal has an implied performance index of 0.81, which means that service classes with discretionary goals will always be considered as achieving their service goal.

[27]Recall that the policy adjustment interval is 10 seconds of elapsed time.

- **OW25542 has been applied**. With OW25542, the Workload Manager will examine the performance of all service class periods at the highest goal importance **using the local performance index** as the measure of performance. The Workload Manager will continue examining performance at successively lower goal importance levels, analyzing performance based on sysplex performance index followed by an analysis of performance based on local performance index.

  Both the original design of the Workload Manager and the fix for OW25542 operate under a basic assumption: that a sysplex consists of multiple systems configured in a symmetric manner, and that service class periods can operate on any system in the sysplex. If the workload being processed consists of transaction service classes such as CICS transactions managed by CICSplex/SM and routed to any system in the sysplex to be processed in cloned CICS regions, this view of the sysplex makes sense.

  From this perspective, all systems in the sysplex can be viewed collectively as a pool of resources and the performance of the transactions can be evaluated based on how well the transactions perform on the sysplex. If a service class period is not meeting its performance goal *on the sysplex*, action may or may not be necessary at a local system level. Consequently, **sysplex performance index** is the basic measure of performance used in the Workload Manager design.

  Unfortunately, this logic does not work in all situations. Consider a site that has established a service class for TSO trivial transactions. The TSO users might log onto, for example, two systems: SYSA and SYST. The users on SYSA might represent production work while the users on SYST might represent TSO testing (and might not be as important to the site as the production work).

The first service class period encountered with a Performance Index greater than one is selected for attention, as a Performance Index greater than one means that the performance goal is not being met. There are two exceptions to this selection:

- The Workload Manager will not attempt to reallocate resources if the Performance Index is only slightly above one[28]. This prevents application of the resource allocation algorithms where only marginal performance improvement is to be expected.

- If a service class cannot be "helped" after the policy adjustment algorithms are executed and analyzed, the Workload Manager will set a "skip counter" for the service class. The skip counter initially is set to three and decremented each policy adjustment interval, until the skip counter reaches zero. While the skip counter for a service class is above zero, the Workload Manager will ignore the service class unless performance significantly degrades.

[28]The actual value used is 1.1 (that is, the Workload Manager will not attempt to reallocate resources if the Performance Index is 1.1 or below).

The reasoning here is that there is little point in continuing to try to help a service class if the first attempt fails. It is better to wait awhile with the expectation that system conditions will change.

The Workload Manager waits 20 seconds (2 policy adjustment intervals) before the service class is again subjected to the policy adjustment algorithms. If performance seriously degrades for the service class, the skip logic is ignored, and the service class will be selected for policy adjustment if it has a sufficiently high importance.

Once a service class has been selected for consideration of policy adjustment, the Workload Manager considers the service class to be a "goal receiver" of resources and begins looking for a service class which can be a "donor" of resources. A potential donor is selected in inverse order by which a receiver is selected (that is, the least important service classes are examined and a service class from that group with the lowest Performance Index is selected as a potential donor).

Within the same Goal Importance, the Workload Manager will not take resources from a service class to help another service class unless the expected performance improvement to the "receiver" service class outweighs the expected performance degradation to the "donor" service class.

A variety of algorithms are applied to assess whether the receiver service class can benefit from additional CPU dispatching priority, additional processor storage, etc. These algorithms are illustrated in IBM's *Programming: Workload Management Services*, Chapter 4 (Using SMF Record Type 99).

Additionally, if the "goal receiver" and potential "donor" are at the same goal importance level, the harm to the donor service class by losing the resources is evaluated. Within a goal importance level, the Workload Manager reallocates resources only if the projected benefit outweighs the projected harm. This evaluation is not done if the potential "donor" is at a lower importance level than the "receiver". The resource allocation is made regardless of the potential harm to the "donor" as the goal importance takes precedence.

If policy adjustment algorithms conclude that resource adjustment would be beneficial, the Workload Manager adjusts the policy by altering the SRM's internal controls so the SRM will manage system resources according to the new controls.

Only **one** receiver is helped during a single policy adjustment interval, although multiple donors may be required to provide the necessary resources. Only one receiver is helped since the Workload Manager must acquire additional data to evaluate whether the particular resources helped performance for the receiver.

Exhibit 4-11 illustrates a sample ordering of service classes within Goal Importance and Performance Index (the service classes are arbitrarily numbered 1-12 for the sake of

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

brevity). This exhibit will be used to illustrate the above concepts, **assuming that OW25542 has not been applied**.

SERVICE CLASS	IMP	SYSPLEX PERF INDEX
1	1	1.1
2	1	0.8
3	1	0.7
4	1	0.6
5	2	4.2
6	2	3.9
7	2	1.0
8	2	0.9
9	2	0.7
10	3	6.0
11	3	5.3
12	3	1.2

SAMPLE ORDERING OF SERVICE CLASSES

Exhibit 4-11

- The Workload Manager will examine the service classes in the order shown (top to bottom) since Service Class 1 has the highest Goal Importance[29] and this service class has the highest Performance Index within Goal Importance 1.

- The Workload Manager takes no action on Service Class 1 since the Performance Index is 1.1, and the Workload Manager does not wish to try to help a service class which is very close to achieving its performance goal.

- The Workload Manager next examines Service Class 2. This service class and all subsequent service classes at Goal Importance 1 are skipped, since the Performance index for these service classes is less than one (indicating that the service classes are achieving their performance goal).

- The Workload Manager will next examine Goal Importance 2, beginning with Service Class 5. This service class has a Performance Index of 4.2, so the Workload Manager

[29]Recall that Goal Importance 1 is the highest, while Goal Importance 5 is the lowest. Goal Importance 6 is used internally by the Workload Manager for discretionary work since discretionary work is the lowest importance of all.

will attempt to help the performance of this service class. Service Class 2 will be selected as a potential "receiver" for resources.

- The Workload Manager will select Service Class 12 as a potential "donor" of resources to Service Class 5. Service Class 12 was selected because it was the service class with the lowest Goal Importance and the lowest Performance Index.

- The Workload Manager will arbitrarily take any necessary resources from Service Class 12 in an attempt to help Service Class 5.

- If Service Class 12 does not hold enough resources to help Service Class 5, the Workload Manager will select Service Class 11 as a potential donor, and then select Service Class 10 as a potential donor.

- Suppose that Service Class 10, Service Class 11, and Service Class 12 do not hold sufficient resources to help Service Class 5. The Workload Manager will then select Service Class 9 as the next potential donor. Service Class 9 would be selected because it was the service with the next lowest Goal Importance and the lowest Performance Index.

- Once the Workload Manager selected Service Class 9, the Workload Manager would take resources from Service Class 9 **only if the projected performance improvement for Service Class 5 exceeds the projected performance degradation to Service Class 9**. The reasoning here is that the service classes are at the same Goal Importance, and resources should be reallocated to service classes at the same importance only if there is a "net gain" for the action.

- Suppose the Workload Manager concluded that the performance of Service Class 5 could not be significantly improved after analyzing Service Class 6 through Service Class 12. The Workload Manager would set a "skip counter" for Service Class 5 and select Service Class 6 as a potential "receiver" of resources. Unless the performance of Service Class 5 **significantly** deteriorated, the Workload Manager would skip Service Class 5 for three subsequent policy adjustment intervals.

  The reasoning here is that (1) Service Class 5 could not be improved with current system conditions and (2) perhaps system conditions would change after some elapsed time. The three policy adjustment intervals equates to an elapsed time of 30 seconds real time.

The Workload Manager may adjust resource allocation policy in the following areas:

- CPU dispatching priorities and CPU capping

- MPL targets for domains (recall that a domain is created for each service class period).

©Copyright 1994, Computer Management Sciences, Inc.

- Swap protect time (the time address spaces will be protected in processor storage before being eligible for swap to auxiliary storage).

- Expanded storage policies (swap working set pages, VIO pages, hiperspace pages, and stolen pages/swap trim pages).

- Protected storage targets (central storage protection, central storage restriction, expanded storage protection, expanded storage restriction).

With OS/390 Release 3, the Workload manager may adjust resource allocation policy in the I/O priority queuing area. I/O priority queuing is used to control non-paging DASD I/O requests that are queued because a device is busy.

- Without I/O priority queue management, the Workload Manager sets I/O priorities equal to the dispatching priority for the service class period.

- With I/O priority queue management, the Workload Manager will dynamically adjust the priority of I/O requests based on how well the service class is meeting its goal, and whether the device can contribute to meeting the goal. The Workload Manager will consider I/O priority queue adjustment only if you have specified *I/O priority management = yes* in the Workload Manager Service Coefficient/Service Definition Options panel.

If policy adjustment algorithms conclude that resource adjustment would not be beneficial, the Workload Manager selects another potential receiver service class (in order of ascending importance and descending Performance Index).

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 4: Workload Manager Performance Data

The primary source of performance information about the performance of service classes is contained in SMF Type 72 (Subtype 3) records[30]. The Type 72 records contain a **wealth** of information, with each record describing a service class period in great detail. The Type 72 records include:

- Workload Manager control information: service policy name and activation time, service coefficients, workload group name, etc. This information is collected from the service policy in effect when the SMF Type 72 record was written.

- Service class served data (for subsystems serving other service classes): the name of each service class served and the number of times the service class was served. This information is collected from the service policy in effect when the SMF Type 72 record was written.

- Resource group information (if a service class is associated with a resource group): resource group name, CPU minimum, and CPU maximum. This information is collected from the service policy in effect when the SMF Type 72 record was written.

- Service class period data is available in several categories

  - Description of service class period: type of performance goal, value of the performance goal, importance of the goal, and performance period duration. This information is collected from the service policy in effect when the SMF Type 72 record was written.

  - Resource data: CPU service units, I/O service units, central storage service units, page-in counts by several categories from auxiliary and expanded, storage residency time, swap count, and service times.

    The resource values are not available for "served" service classes (e.g., CICS transactions), but are available for the "server" subsystems (e.g., the CICS regions serving the transactions). Please refer to Chapter 2 of this section for discussion of the served and server service classes.

  - Transaction data: transaction elapsed and execution times, and transaction ending counts.

[30]The SMF Type 99 records contain extremely detailed information about the internal logic and decisions of the Workload Manager. The Workload Manager creates a "trace" of its logic as it examines service classes, makes resource adjustment decisions, makes policy adjustment decisions, working set management decisions, etc. The Type 99 records are written every 10 seconds, and may be extremely large. IBM strongly recommends that Type 99 records be collected for **only** a short time in response to specific needs. Since the Type 99 records are not normally collected, CPExpert does not consider them to be a primary source of performance data. In contrast, the Type 72 records are routinely collected at virtually every site.

©Copyright 1994, Computer Management Sciences, Inc.

- Samples describing address space states: CPU using samples, execution delay samples, and other samples related to address spaces in the service class.

- Response time distribution data (for service classes with response goals): a distribution of response time into 14 counters of transactions ending with response times relative to the performance goal.

  The following table shows the percentages as a function of the response goal, recorded into each counter. Each percentage reflected in the table describes a "counter" recorded by RMF in SMF Type 72 records, and reflects a count of the transactions with response times as the indicated percentage of the response goal. For example, a count of "13" in the second counter (50%-60%) would indicate that 13 transactions ended with a response time of between 50% and 60% of the response goal.

COUNTER GOAL	PERCENT OF
1	LESS THAN 50%
2	50%-60%
3	60%-70%
4	70%-80%
5	80%-90%
6	90%-100%
7	100%-110%
8	110%-120%
9	120%-130%
10	130%-140%
11	140%-150%
12	150%-200%
13	200%-400%
14	OVER 400%

RESPONSE TIME DISTRIBUTIONS
SMF TYPE 72 RECORDS

Exhibit 4-12

The last counter contains a count of transactions which exceeded 400% of the performance goal.

Additionally, for subsystems supporting Workload Manager interface (initially, CICS/ESA Version 4.1 and IMS Version 5), the following work manager/resource manager information is available in the Type 72 records:

- Work Manager/Resource begin_to_end phase state samples: active state, ready state, idle state, waiting states, and switched states.

- Work Manager/Resource execution phase state samples: active state, ready state, idle state, waiting states, and continuation states.

If transactions in the service class are served by both CICS and IMS, the Type 72 records contain begin_to_end phase and execution phase information for CICS and only execution phase information for IMS.

These subsystem states are described in Chapter 2 of this section.

©Copyright 1994, Computer Management Sciences, Inc.

Chapter 5: RMF Data Analysis Considerations

This chapter highlights some of the factors that must be considered when analyzing workload data collected and recorded by RMF in SMF Type 72 records.

These factors do **not** preclude a comprehensive analysis of performance data and usually do not prevent insight into the causes of unacceptable performance. However, the factors must be recognized and accounted for both by CPExpert in analyzing data and by the user in reviewing CPExpert's results. The factors stem from (1) the way in which the SRM defines transactions, (2) the way in which RMF collects transaction counts and times, and (3) operator actions which cause unique data recording.

Chapter 5.1: SRM Transactions

Within MVS, the concept of a "transaction" is based upon the SRM's definition of a transaction. This definition differs depending upon whether the transaction statistics relate to TSO, relate to batch, or relate to a started task (such as CICS). The Workload Manager uses the SRM services to acquire information and modifies SRM control blocks to adjust system policies. Consequently, the SRM's definition of a transaction is relevant to understanding RMF data.

- TSO transactions begin (1) whenever terminal input is entered and the line is not continued, (2) the 3270 field mark key separates commands on the same input line, (3) a command's output is detained while waiting for an output buffer, or (4) a command is taken from the TSO internal stack.

- Batch transactions correspond to a job in most cases. Batch jobs are single transactions counted in their service class, unless they transition to a different performance period. In this case, the batch transaction is "ended" and a new transaction begins with the new step and the new performance group.

- Most started tasks are counted as a single transaction for their entire execution time (unless they employ special techniques to communicate "transaction" information to the SRM, or unless SRM counters are in danger of overflowing). Consequently, individual transactions in support of started tasks are not managed or counted by the SRM[31].

For example, CICS transactions are managed by CICS, rather than the SRM. The SRM views the entire CICS region as a single transaction that ends when the CICS region ends.

[31]With CICS/ESA Version 4.1, CICS communicates some transaction information to the Workload Manager if the CICS transactions have been placed into their own service class. Similarly, with IMS/ESA Version 5, IMS communicates some transaction information to the Workload Manager. Consequently, the Workload Manager has some information about transactions, but the SRM does not collect resource use at the transaction level, as the SRM manages and collects information on address spaces.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

With versions of MVS prior to MVS/ESA SP5.1 (Goal Mode), it was convenient to think of a transaction as being synonymous with an "address space" since only one transaction (from the SRM's view) could be active in an address space at one time. Of course, subsystems (e.g., CICS, DB2, IMS) could process many transactions, but as explained above, the SRM would view the subsystem as a single transaction.

With MVS/ESA SP5.1, the SRM still views the address space as a single transaction. This view is still relevant for MVS components except subsystems (that is, the view is relevant for TSO transactions, batch jobs, etc.). RMF records transaction information from the SRM's view (e.g., transaction counts, transaction active time, transaction elapsed time, etc.).

With MVS/ESA SP5 (Goal Mode), subsystems can communicate transaction information to the Workload Manager via Workload Manager Services macros. These transactions are individual CICS transactions and individual IMS transactions. The Workload Manager accumulates transaction information for the subsystems, and RMF records the subsystem information in SMF Type 72 (Subtype 3) *Work Manager/Resource Manager Section*. The Work Manager/Resource Manager data will be discussed later in Section 5.4. Initially, we will restrict our discussion to transactions from the SRM's view.

A transaction becomes "ready" from the SRM's view when the transaction starts. Transaction active time and transaction elapsed time begin at this point.

The address space supporting a newly-ready transaction is in a swapped out state (except for CICS or IMS regions if they have been made non-swappable). The address space must be brought into the system and made a part of the multiprogramming set[32]. The address space may remain swapped out for some period of time before being swapped into storage.

After some swap delay, the address space supporting the transaction is made resident in storage (or the address space supporting the transaction's control blocks are updated, in the case of logically swapped address spaces). Once resident, the address space may be swapped out of storage for a large number of reasons (these reasons are globally recorded in SMF Type 71 records and include exchange swaps, unilateral swaps, etc.). If the swap was uncontrolled by the address space, the time is counted as transaction active time (the address space is still ready to execute).

Thus, the **transaction active time** is the address space resident time plus the time the address space was swapped out but ready to execute. The **transaction elapsed time** is the total time from transaction start to transaction end, regardless of whether the address space was ready to execute.

[32]This discussion ignores the implications of logical swapping, in which the transaction is submitted by an interactive user, and the address space may actually reside in storage even though "logically" swapped out.

©Copyright 1994, Computer Management Sciences, Inc.

All transactions are counted by the SRM when they end, with the above definition of "ending" transactions. Transaction counts and the elapsed times of transactions are recorded **when the transaction ends**. This method of recording can have a profound impact on transaction counts and times as recorded by RMF, and on averages based upon these transaction counts and times.

Chapter 5.2: RMF Transaction Information

Transaction-related information is acquired by RMF at the end of a recording interval. The information is recorded in SMF Type 72 (Subtype 3) records. RMF **normally** records this information at user-specified intervals (e.g., users commonly specify RMF recording every 15 or 30 minutes)[33].

There are two implications with the information recorded into SMF Type 72 records: (1) RMF normally collects and records information at regular intervals but (2) the transaction counts and times are available to RMF only when transactions end. These two facts can distort all computations based upon transaction counts or times.

- **Average response time**. Most analysts compute average response time by dividing the transaction elapsed time (SMF72TTM) by the transaction count (SMF72TTX). These SMF variables are applicable for MVS prior to Goal Mode; with Goal Mode, the variables are R723CTET and R723CRCP, respectively. These are the values used to compute response time in the RMF *Workload Activity Report*.

  The transaction elapsed time represents the entire transaction elapsed time, including long wait (e.g., time enqueued, waiting for a response to WTOR, etc.). The long wait part of some transactions can be extremely lengthy. The lengthy elapsed times of only a single transaction can completely distort the average transaction time. In fact, transactions can be in a long wait condition for many minutes. In some situations[34], the average response resulting from using SMF72TTM (or R723CTET) can exceed the RMF recording interval in which the transactions finally end!

  This situation is not usually serious for the computing the average response times of a TSO Period 1 service class. TSO Period 1 is intended to service interactive transactions. These transactions are very short relative to the RMF recording interval, and have less of an opportunity to "spill over" into a RMF recording interval different from which they started. Additionally, there usually is a large number of transactions ending in TSO Period 1 for any RMF interval. Thus, there is a greater chance of "evening out" the average number transactions ending.

[33]There are important exceptions to this normal recording, as discussed later.

[34]For example, such distortions might occur if TSO were used by a single operator on the midnight shift, and extensive Enqueue time was experienced.

This same reasoning normally applies to TSO Period 2, although more care must be taken with the results.

The situation becomes more serious for TSO Period 3 or Period 4 transactions and for batch jobs. The results can be seriously misleading if average resource per transaction is computed for these workloads based upon SMF Type 72 workload information.

Fortunately, IBM provides **response time distributions relative to the performance goal** for all service class periods which have response performance goals. Thus, although average response time may not be particularly meaningful, the response time distributions can be analyzed to understand the response experienced by the service class period.

- **Average resources used**. The resources used by transactions (e.g., CPU service units, memory service units, etc.) and the resource counts (e.g., swap sequences and page-ins) are **continually accumulated**. This information is acquired by RMF in the recording interval **in which the resources were used**.

  In computing the average resource per transaction, the resource is divided by the count of ended transactions (SMF72TTX or R723CTET). As explained above, the count does not necessarily represent the transactions actually using resources. There could be many transactions (e.g., long- running batch jobs) in the multiprogramming set. These transactions could use significant resources, but their count would not be reflected in computing the average resources used unless they ended in the RMF recording interval.

  There are several additional scenarios that complicate the average resource computations:

  Resources are accumulated in the performance **period in which they are used**. Transactions are counted in the performance **period in which they end**. For example, the resources required to support the TSO service class Period 2 transactions while they were in Period 1 are attributed to Period 1.

  However, the TSO Period 2 transactions are not counted in Period 1; they are counted in Period 2 if they end in Period 2 (however, if they migrate to Period 3, they are not counted in Period 2 but would be counted in Period 3 if they end in Period 3). This situation has the effect of inflating average resources used in Period 1 by whatever was used by transactions ending in Period 2. (Of course, the same applies for resources used by transactions ending in Period 3 and Period 4.)

  If numerous transactions pass through Period 1 to (for example) Period 4 within a single RMF measurement interval, the average service rate for Period 1 transactions could be extremely large. In fact, the average service rate could be far more than the

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

computer system was able to deliver! Such data anomalies can give analysts concern if they are not aware of the causes.

It is possible to "back out" some of this resource use by estimating the resources used by TSO transactions ending in Period 2, Period 3, etc. However, there is no guarantee that these transactions will end in the same RMF measurement interval, and thus no knowledge of how much should be "backed out" of the resources.

CPExpert exploits this situation for service classes having multiple periods and having a relatively short response goal for Period 1. CPExpert analyzes the average service used in Period 1 compared to the DUR value for Period 1. If the average service used is much greater than the DUR value, conclusions can be made about non-interactive transactions executing in Period 1.

Chapter 5.3: Operator Actions

An additional problematic situation occurs if the service policy is changed (even if the service policy is changed simply to implement policy overrides)[35]. When the service policy is changed, RMF writes out SMF Type 72 records before the new service policy takes effect.

When the service policy is changed, all ongoing transactions are ended for any service class effected, and the transaction is restarted in the first period of their service class.

RMF collects workload and resource information before writing out its records. Therefore, changing the service policy could have the effect of creating a large number of "ended" transactions in a short RMF recording interval. This large number would seriously distort the workload averages computed for the period.

After writing out the information acquired before the new service policy takes effect, RMF will synchronize the writing of SMF Type 72 records with the writing of other system records at the user specified interval (e.g., every 15 minutes). Thus, if the service policy is changed, there will be two SMF Type 72 records written for the RMF recording interval in which the service policy change occurred. Either of these SMF Type 72 records could represent a **very** short interval if the policy change occurred just after the previous recording interval or just before the current recording interval.

There potentially is a serious performance implication of changing the service policy. The Workload Manager will require some elapsed time to accumulate statistics about the performance of service classes in the new service policy. Depending upon the number of

[35]The same effects occur if a mode change is made (switching between Compatibility Mode and Goal Mode). This discussion would become cluttered if all change references used "service policy change or mode change" to describe the effects. Of course, the effect would be more serious if a mode change were made. However, mode changes are likely to be infrequent once users are comfortable with Goal Mode operation.

service classes and types of performance goals, some significant elapsed time could be required for the Workload Manager to adjust resource allocation to the new service policy.

Additionally, all ended transactions would restart in Period 1 of their service class. This could have the effect of suddenly placing non-interactive transactions into Period 1 of the service class, and these transactions would compete with interactive transactions for system resources. As a result, there could be some interval of poor response to the interactive users.

Chapter 5.4: SRM Sampling of Subsystem State

CICS/ESA Version 4.1 or IMS Version 5 will provide the Workload Manager with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The Workload Manager simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[36], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class.

The SRM also keeps a count of the number of samples which it takes of the begin_to_end phase and of the execution phase.

The counts of various samples are recorded in the "Work Manager/Resource Manager State Section" of SMF Type 72 records.

The SRM also includes the elapsed time of transactions (R723CTET) and the count of transactions (R723CRCP) in the SMF Type 72 records. Based on the transaction elapsed time and transaction count, CPExpert can compute the approximate number of samples which the SRM **should** take of the begin_to_end phase of transactions.

To illustrate the computation, suppose that a single transaction were to execute in a service class, and further suppose that the transaction elapsed time was 1 second. During this second of elapsed time, the SRM should take a sample every 250 milliseconds (4 samples per second), or 4 samples of the begin_to_end phase[37] of the transaction of the 1-second transaction. If two transactions with individual elapsed times of 1 second were

[36]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

[37]For the moment, we can ignore the time required by MVS to assign the transaction to a CICS region, the time for the CICS region to issue the IWMCLSFY macro, the time for the Workload Manager to classify the transaction to a service class, and the time for the CICS TOR to issue the IWWMINIT macro. These times normally are very small.

©Copyright 1994, Computer Management Sciences, Inc. Revised: November, 2001

to execute in the service class, the SRM should take 8 samples (1 second average elapsed time * 2 transactions * 4 samples per second = 8).

Thus, the computation of the number of samples which the SRM **should** take in any RMF measurement interval is simply the total elapsed time of transactions, times the sampling rate. The result from this computation should always be more than the number of samples which the SRM actually took of the begin_to_end phase, since the begin_to_end phase does not start until after the transaction has entered the system and has been classified to a service class, and the begin_to_end phase ends before the transaction is finally marked "ended" by the SRM.

However, the SRM updates the elapsed time of transactions only when the transactions end. Suppose that a never-ending transaction executed in the service class. The SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase). However, the SRM would never see the transaction complete and thus would not update the elapsed time of the transaction.

A similar situation occurs with long-running transactions. These transactions can span RMF measurement intervals; the SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase) in one RMF interval. The elapsed time of the transaction might not be recorded until a subsequent RMF interval.

These anomalies can cause response time calculations to be misleading. More importantly, the Workload Manager algorithms may be less effective if never-ending or long-running transactions are in the same service class as interactive transactions. This is because the Workload Manager's computation of response times may be distorted by the long-running transactions.

CPExpert detects many of these situations and either adjusts its analysis of performance problems or produces rules to alert you to the problems.

For example, Rule WLM110 to Rule WLM113 describe the results of CPExpert's analysis to detect never-ending or long running transactions executing in a service class with interactive transactions.

©Copyright 1994, Computer Management Sciences, Inc.

Description of Rules

Appendix A

This appendix contains a description of each rule that results in a finding by the WLM Component of CPExpert. The description summarizes the rule, lists predecessor rules, discusses the rationale for the finding, and suggests action. The Appendix is contained in both Volume 1 and Volume 2 of this User Manual.

The summary of the rule presents a short description of the finding.

The predecessor rules are listed so you can follow the line of reasoning leading to a particular rule being executed.

The discussion describes as much as necessary of the operation of the computer system (the hardware, the WLM, the SRM, etc.) as it relates to the particular rule. The purpose of the discussion is to explain the reasoning behind the rule, and what causes the rule to be produced.

The suggestions list possible actions that should be considered based on the findings. In many cases, multiple possible actions are listed. You must determine which actions should be taken (this determination is based upon the suitability of the actions to your own environment, the financial implications of the action, and the "political" acceptability of the action.)

The rules are organized in numerical order. However, not all numbers are represented (for example, RULE WLM200 follows RULE WLM150). The LIST OF RULES in this appendix lists all rules that are included in the current release of the WLM Component. Within the rule framework, the following general categories apply:

- **Service Policy Findings**. The Service Policy Findings are rules in the WLM001 to WLM050 range. These findings help identify problems or potential problems with the Workload Manager service definition. The Service Policy Findings are contained in Volume 1.

  It is important to realize that these findings normally identify a POTENTIAL problem. Your systems programming staff must decide whether the findings (and their associated recommendations) make sense in your environment. For example, your systems programming staff might have deliberately selected certain parameter values. The values might be appropriate for your installation and your management objectives, even though CPExpert might produce a rule indicating that there is a potential problem with the parameter.

  You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is set to N, CPExpert will not check the service definition for potential problems.

©Copyright 1994, Computer Management Sciences, Inc.

- **General System Findings**. The General System Findings are rules in the WLM050 to WLM099 range. These findings identify problems or potential problems with your overall system. For example, many of the rules deal with problems with the paging subsystem. These findings are made only if CPExpert detected that a performance goal was not met and that some general system problem might have caused the goal to be missed. The General System Findings are contained in Volume 1.

- **Specific Findings**. The Specific Findings are rules above WLM100. These findings are made if CPExpert detected that a service class did not meet its performance goal. In the Specific Findings, CPExpert attempts to isolate the reason(s) the performance goal was not met. The Specific Findings are contained in Volume 2.

  WLM1nn(series) relate to performance goal findings

  WLM2nn(series) relate to CPU-related findings

  WLM3nn(series) relate to UNKNOWN delay findings

  WLM4nn(series) relate to swap-in and target MPL findings

  WLM6nn(series) relate to Cross System Coupling Facility (XCF) findings

  WLM7nn(series) relate to System Logger findings

You may wish to read all of the rules in this appendix, just to see the type of problems that are encountered in different installations. However, it is not necessary to read all of the rules. It is necessary only to read the rules that apply to your installation. The rules that apply to your installation are identified by the report produced from the WLMCPE Module.

All references to *MVS Initialization and Tuning Guides* or *MVS Initialization and Tuning References* apply to the following specific documents:

MVS/XA Initialization and Tuning Guide, GC28-1149-4

MVS/ESA SP3.1 Initialization and Tuning Guide, GC28-1828-2

MVS/ESA SP4.1 Initialization and Tuning Guide, GC28-1634
MVS/ESA SP4.1 Initialization and Tuning Reference, GC28-1635

MVS/ESA SP4.2 Initialization and Tuning Guide, GC28-1634-3
MVS/ESA SP4.2 Initialization and Tuning Reference, GC28-1635-3

MVS/ESA SP4.3 Initialization and Tuning Guide, GC28-1634-4

MVS/ESA SP4.3 Initialization and Tuning Reference, GC28-1635-4

IBM released a new version of the *Initialization and Tuning Guide* and *Initialization and Tuning Reference* for SP4.3 in January 1994. The following documents are used for references updated after January 1994.

MVS/ESA SP4.3 Initialization and Tuning Guide, GC28-1634-5
MVS/ESA SP4.3 Initialization and Tuning Reference, GC28-1635-5

Beginning with MVS/ESA SP5.1, the references to IBM documents **apply to IBM BookManager documents**. This change was made because all CPExpert users installing MVS/ESA SP5.1 also use IBM BookManager to access soft-copy IBM documents rather than acquiring hard-copy IBM documents.

- The IBM BookManager documents are contained in IBM CDROM LK2T-5114 or in IBM CDROM SK2T-0710 (with appropriate quarterly updates).
- With OS/390, the IBM BookManager documents are contained in IBM CDROM SK2T-6700.
- With z/OS, the IBM BookManager documents are contained in IBM CDROM SK3T-4269.

If any user does not have access to IBM BookManager documents, please call Computer Management Sciences. We will be happy to provide references to hardcopy manuals.

Beginning with CICS/Transaction Server for z/OS, CICS documentation is contained in the CICS Information Center (InfoCenter). IBM provides the following description of the documentation available with CICS/Transaction Server for z/OS:

"For CICS Transaction Server V2.1 (announced March 2001), there has been a move away from printed books as the default deliverable to a new online concept. The primary source of user information for this release is a new CICS Information Center with a graphical user interface, delivered with the product on a CD-ROM. This HTML-based Information Center runs inside a Web browser, and provides a number of alternative means of accessing the information within it.

The objective of the Information Center is to make it easy for users to retrieve the information they need to perform specific CICS tasks, or to find relevant background or reference information on demand. At the heart of the Information Center is an HTML representation of the total CICS library (unlicensed books) Within the graphical user interface, the key documentation can be accessed via three main classes: tasks, concepts, and reference, each separately selectable. On selecting a class, the categories for that class are displayed in the navigation panel. Each of these can be expanded into a hierarchical navigation tree of topics in turn point to the detailed information.

The Information Center also includes a powerful search capability based on IBM's NetQuestion technology. Search results can be saved for future reference. In addition to the new methods of accessing the CICS documentation, the Information Center provides the more traditional alternative of a complete library listing of the books, which can be viewed in both HTML and PDF formats. The latter also provides the capability to print either the whole book or some of the pages in hardcopy a printer, using Adobe Acrobat.

For this new release of CICS, the main focus of the documentation is the implementation of EJB technology in the CICS environment. A new piece of documentation entitled "Java Applications in CICS" is the cornerstone of this information, and has been designed to make use of the new capabilities of the Information Center."

CPExpert references for CICS/Transaction Server for z/OS are specific to the CICS Information Center.

List of Rules
Volume 1

RULE	DESCRIPTION
WLM001	The service class definition may not match workload
WLM002	Conflict exists between service class and report class
WLM003	The service policy was changed
WLM004	CPExpert believes too many service policy changes occurred
WLM005	The velocity goal may be too high for batch service class
WLM006	The response time goal is too large
WLM007	MSO service definition coefficient may be too large
WLM008	DUR value may be too large for TSO Period 1
WLM009	Minimum CPU service specified for Resource Group
WLM010	Velocity goals have values which are too similar
WLM011	The service definition does not describe all workloads
WLM012	A server workload defaulted to the SYSSTC service class
WLM013	Response goal was specified for a server service class
WLM014	Response goal specified for "hot batch" workload
WLM015	Execution velocity goal specified for TSO Period 1 or Period 2
WLM016	Low execution velocity goal specified for server service class
WLM017	Server and subsystem transactions in same service class
WLM018	Multiple periods specified for server service class

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 1

RULE	DESCRIPTION
WLM019	Multiple periods specified for subsystem transaction service class
WLM020	Subsystem transactions in same service class as address space
WLM021	Subsystem transactions service class assigned to resource group
WLM022	Execution velocity goal specified for subsystem transaction service class
WLM023	Too many service class periods may have been specified
WLM024	More than three periods were specified for a service class
WLM025	The service class period may be unnecessary
WLM026	Highest importance service class period had few samples
WLM027	Service class periods have same velocity goal and importance
WLM030	Report class period is heterogeneous
WLM031	Dynamic alias management was active but I/O priority management was not selected.
WLM032	Server was assigned CPU protection, but most work was done in support of lower importance work

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 1

RULE	DESCRIPTION
WLM050	The number of available page slots is low
WLM051	The number of local page data sets may be inadequate
WLM052	The number of allocated page slots may be insufficient
WLM053	The number of allocated page slots may be insufficient
WLM054	The number of allocated page slots may be insufficient
WLM055	Local page data sets are on same volume as swap data sets
WLM056	Local page data sets share volume with COMMON or PLPA
WLM057	Multiple local page data sets are on the same volume
WLM058	Local page response is significantly worse than average
WLM059	Insufficient local page data sets are defined for migration
WLM060	PLPA and COMMON page data sets may be combined
WLM061	Swap data sets are defined
WLM070	Terminal Output Wait swaps occur too often
WLM071	Detected Wait swaps occur too often
WLM080	JES-managed and WLM-managed job classes conflict
WLM081	WLM-managed job class assigned to multiple service classes
WLM082	Job might not be suitable for WLM-managed initiators
WLM090	SMF Type 30 interval recording not turned on

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM101	Service class did not achieve average response goal
WLM102	Service class did not achieve percentile response goal
WLM103	Service class did not achieve velocity goal
WLM104	Served service class did not achieve average response goal
WLM105	Served service class did not achieve percentile response goal
WLM106	Response time distribution for service class
WLM107	Response time distribution for service class
WLM108	Response time distribution for served service class
WLM109	Response time distribution for served service class
WLM110	BTE Phase samples count was larger than calculated samples
WLM111	BTE Phase Idle sample count is large
WLM112	BTE Phase had large (Ready plus Active) sample count
WLM113	BTE sample count was significantly less than calculated samples
WLM114	BTE phase had large ready samples
WLM115	Service class did not have begin_to_end phase samples
WLM116	Execution Phase samples did not exist in SMF data
WLM117	Transaction service class wait states
WLM119	Work manager data was not collected for service class

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM120	Significant transaction time was in Active state
WLM121	Significant transaction time was in Ready state
WLM122	Significant transaction time was in Idle state
WLM123	Significant transaction time was Waiting for Lock
WLM124	Significant transaction time was Waiting for I/O request
WLM125	Significant transaction time was Waiting for Conversation
WLM126	Significant transaction time was Waiting, Distributed
WLM127	Significant transaction time was Waiting, Local Session
WLM128	Significant transaction time was Waiting, Sysplex Session
WLM129	Significant transaction time was Waiting, Network Session
WLM130	Significant transaction time was Waiting for Timer
WLM131	Significant transaction time was Waiting, Another Product
WLM132	Significant transaction time was Waiting, Miscellaneous
WLM135	IMS activity processing transactions in service class
WLM136	DB2 activity processing transactions in service class
WLM140	Sysplex performance index was significantly less than local
WLM150	Server service class delays (single transaction service class)
WLM151	Server service class delays (multiple transaction service classes)
WLM152	Server served multiple transaction service classes

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM153	Server served multiple transaction service classes
WLM170	Address spaces were idle a significant percent of time
WLM171	Execution velocity was based on a small sample set
WLM172	Server was idle a significant percent of time
WLM173	The response performance goal may be too large

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM200	Average CPU use per transaction is higher than goal
WLM201	Goal may be unrealistic - average CPU use is high
WLM202	Average CPU use was a major cause of transaction delay
WLM210	Average server CPU use per transaction is higher than goal
WLM211	Goal may be unrealistic - average server CPU use is high
WLM212	Average CPU use was a major cause of transaction delay
WLM220	Service class was delayed because of resource capping
WLM221	Service Class was capped for discretionary goal management
WLM222	Service class was Active, but server was CPU capped
WLM250	Service class waited for access to CPU
WLM251	Dispatcher reduced preemption might have caused CPU delay
WLM252	CPU access might be denied because of Resource Group minimum
WLM255	Service class was active but server was denied CPU
WLM256	Service class was active and server was not denied CPU

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM340	Batch jobs may be delayed waiting for an initiator
WLM341	Service class may be waiting for initiator/scheduler
WLM350	I/O activity may have caused significant delays
WLM351	I/O activity may have caused significant delays
WLM352	I/O activity may have caused significant delays to server
WLM353	I/O activity may have caused significant delays to server
WLM355	Device DISConnect time was a major cause of DASD delays
WLM356	Device PEND time was a major cause of DASD delays
WLM357	Device CONNect time was a major cause of DASD delays
WLM358	Device IOS queuing time was a major cause of DASD delays
WLM359	I/O activity probably did not cause major delays
WLM360	Service class did not reference DASD

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM361	Non-paging DASD I/O activity caused significant delays
WLM362	Non-paging DASD I/O activity caused significant delays
WLM363	Non-paging DASD wait time was a major cause of DASD delays
WLM364	non-paging DASD CONNect time was a major cause of delays
WLM365	Non-paging DASD DISConnect time was a major cause of delays
WLM366	Non-paging DASD IOSQ time was a major cause of DASD delay
WLM370	Non-DASD I/O activtity or delay was significant
WLM371	Non-paging DASD I/O activity caused significant delays
WLM385	SMF Type 30 (Interval) data was not available for service class
WLM390	UNKNOWN delay was not accounted for by above analysis

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM400	Page-in from auxiliary storage was major performance problem
WLM410	Some higher importance service class has storage protection
WLM420	Some equal importance service class has storage protection
WLM450	Swap-in delay was major performance problem
WLM480	Target multiprogramming level delay was major performance problem

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM601	XCF transport class may need to be split
WLM602	XCF message buffer length may be too small
WLM603	XCF message buffer length may be too large
WLM604	XCF outbound message buffer space may be too small
WLM605	XCF inbound message buffer space may be too small
WLM606	XCF local message buffer space may be too small
WLM607	Insufficient outbound paths were defined
WLM608	Transport class did not have a signalling path assigned
WLM620	Message buffer space may be too small for inbound path
WLM621	Message buffer space may be too small for inbound path
WLM622	The number of outbound paths may need to be increased
WLM623	The number of outbound paths may need to be increased
WLM630	A hardware problem may exist
WLM632	An inbound path was non-operational
WLM633	An outbound path was non-operational

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM651	Lock contention was high
WLM652	False lock contention was high
WLM660	Service time was high for synchronous requests
WLM661	Service time was high for asynchronous requests
WLM662	Subchannel contention was high for synchronous requests
WLM665	Too many synchronous requests were changed to asynchronous

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM701	Log stream coupling facility structure was full
WLM702	Log stream staging data set was full
WLM703	Log stream structure offloads occurred: 90% full
WLM704	Interim storage was not efficiently used for log stream
WLM705	Local storage buffers not efficiently used, DASD-only log stream
WLM706	DASD staging data set high threshold was reached
WLM707	Frequent log stream DASD-shifts occurred
WLM708	Log stream caused structure to reach high threshold
WLM709	Log stream consumed most of structure resources

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM001: The service class definition may not match workload

Finding: CPExpert believes that the service class definition may not match the workload executing under control of the Workload Manager.

Impact: This finding should be viewed as a HIGH IMPACT on the performance of your computer system.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: Users define service classes by specifying the workload associated with the service class, specifying the performance goal for the service class, and specifying the importance of the performance goal. Optionally, users may define performance periods for the service class and may assign a resource group to the service class.

Service classes are used by the Workload Manager to determine whether performance goals are being met and to control the allocation of system resources to address spaces. A single transaction, job, Started Task, etc. can be controlled by only one service class (defined in the service policy).

The Workload Manager can assign and reassign system resources only to the extent that system resources exist. The Workload Manager may not be able to manage the system to meet performance goals if (1) the goals are too restrictive, (2) too many goals are at a high importance level, (3) the workload is not sufficiently partitioned into service classes, or (4) the system is inadequate to meet management objectives for the workload being processed.

- **Goals too restrictive**. Performance goals may be defined such that the goals are too restrictive with respect to the resource demands of the workload. For example, Rule WLM200 describes a situation in which the CPU requirements of the average transaction is higher than the performance goal specified for the service class to which the transactions are assigned. This situation commonly occurs when users have defined

goals based on prior performance (for example, based on the *RMF Workload Activity Report* in Compatibility Mode), rather than defining the goals based on management requirements. The Workload Manager may not be able to achieve prior performance for all service classes, but management often does not **require** the prior performance.

- **Too many goals are at high importance**. The Workload Manager attempts to balance the performance goals against resource requirements, considering the importance of the goals. If too many goals are defined at a high level, the Workload Manager may be unable to balance the resource requirements of the service classes. This situation often occurs when performance goals have been assigned to most of the workload, rather than associating a discretionary goal for a significant part of the workload.

- **Workload not sufficiently partitioned**. Implicit in the service class concept is that a service class normally describes a homogeneous workload--a workload which has similar performance goals, similar goal importance, and similar resource requirements. If a service class is defined for workloads which are not similar, the Workload Manager may unable to allocate resources to meet widely-varying resource requirements of the workload.

- **System is inadequate**. The system which is processing the workload may be inadequate to satisfy the performance goals of the service classes. The Workload Manager cannot allocate resources to meet performance goals if insufficient resources exist.

CPExpert concludes that the service class definition does not match the workload if the performance goals with the highest importance levels were not met in most of the RMF measurement intervals being analyzed. CPExpert identifies the highest two goal importance levels. For each RMF interval, CPExpert counts the service classes at these goal importance levels which missed their performance goal. CPExpert produces Rule WLM001 if at least two service classes with these important performance goals missed their performance goal in over half of the RMF measurement intervals being analyzed.

The following example illustrates the output from Rule WLM001:

```
RULE WLM001:  THE SERVICE CLASS DEFINITION MAY NOT MATCH WORKLOAD

  CPExpert believes that you may have a "mismatch" between the service
  classes you have defined and the workload executing on your system, or
  your system is insufficient for the workload.  Perhaps you have defined
  too many service classes at high importance levels, perhaps the goals
  are too restrictive, or perhaps you have not sufficiently partitioned
  your workload into service classes.  Service classes at the highest two
  importance levels did not meet their performance goals during 83% of the
  measurement intervals being analyzed.  When service classes do not meet
  their performance goals, the Workload Manager will attempt to allocate
  system resources so the performance goals can be met.  Since the
  Workload Manager was not successful in doing this in your environment,
  you may wish to review the definition of your service classes and the
  workload assigned.
```

Suggestion: CPExpert suggests that you review the definition of your service classes, the workload assigned to the service classes, the performance goals for the service classes, and the importance of the performance goals. Since the Workload Manager was unable to manage system resources to meet your performance goals for the workload, some changes may be required.

- **Workload classification**. Work units are assigned to service classes by the workload classification scheme. You may have inadequately specified the workload classification scheme (with corresponding assignment of work units to service classes). The result may be that the workloads conflict with respect to their resource demands as matched against the performance goals.

  Many of the rules CPExpert produces suggest that the workload/service class scheme be reviewed. The context of the suggestions (that is, the problem which caused the rule to be produced) should guide you in selecting a different classification scheme if the scheme needs to be revised.

- **Service class definition**. The service class definition often is a trade-off between defining too few service classes and defining too many service classes.

  - **Defining too few service classes**. A service class normally should describe a workload consisting of applications or transactions with roughly the same performance goals, goal importance, and resource demands. Resources are allocated to service classes, based on service class performance goals and goal importance[1]. If the service classes do not adequate discriminate among the applications and

[1]Some Workload Manager actions may be at the address space level. For example, expanded storage policies are different depending upon the performance goal duration. The Workload Manager applies expanded storage policies at the service class level if the performance goal is 20 seconds or less. The Workload Manager applies expanded storage policies at the address space level if the performance goal is greater than 20 seconds.

transactions, the Workload Manager may not be able to allocate resources to meet the varying resource demands.

- **Defining too many service classes**. The Workload Manager will adjust system resource allocation in an attempt to improve performance of only one service class during a policy adjustment interval[2]. Adjustment to improve performance of only one service class is done because the Workload Manager must observe the results of the adjustment; whether the adjustment helped performance, hurt performance, or had no effect. If adjustments were made to improve performance of more than one service class, it would be impossible to determine which adjustment helped or hurt which service class. Consequently, actions to improve the performance of only one service class is done during each policy adjustment interval.

  The policy adjustment interval is 10 seconds. If too many service classes have been defined, the Workload Manager may be able to adjust system resource allocation to help only a few service classes (the most important service classes with the worst performance). Performance of other service classes may never be improved, or performance improvement actions may take a long time - simply because of the elapsed time necessary for the Workload Manager to make changes, collect data, analyze the effect of the changes, make additional changes, collect more data, and continue the process.

  Perhaps of equal significance is the overhead associated with analyzing service classes. The Workload Manager tries to improve performance of the service class with the worst performance at the highest goal importance. Resources may be taken from the least important service class with the best performance. The Workload Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

  The Workload Manager will not add resources unless there is an appreciable net gain to the service class receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class unless the net gain to the receiver outweighs the net loss to the service class the resources are being removed from. The overhead involved with the analysis and decision process increases as the number of service classes becomes large.

[2]The Workload Manager will consider adjustments to improve performance of several service classes (starting with the most important service class which has the highest Performance Index). If performance of the first service class analyzed cannot be appreciably improved, the Workload Manager will select the next worst performing service class, etc. After the Workload Manager has "committed" to a policy adjustment for a service class, it will stop analysis and adjust resources for no other service class.

- **Performance goals**. The performance goals specified for service classes may be too restrictive or they may be specified as averages (rather than percentiles).

  - If the performance goals are too restrictive, the Workload Manager may be unable to allocate system resources to meet the goals. You should specify performance goals with values required to meet management objectives.

    You should not be unduly influenced by pre-Goal mode performance of a particular class of work. Rather, you should establish goals based on organization needs. It is possible that previous performance of certain workloads was at the expense of other workloads or at the expense of relatively poor system utilization.

    The Workload Manager will attempt to achieve high system utilization while meeting performance goals. If the goals are too restrictive, the Workload Manager may have to sacrifice overall system performance to achieve the restrictive goals of the most important service classes.

  - With an average response goal, poor response for only a few transactions can cause the average response goal to be missed. This is because the response time of a few transactions can be **extremely** long. These long response times could cause the average response to be longer than the performance goal, but the resulting average response would not be representative of the performance experienced by the typical transaction.

    With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

    You should use percentile response goals (rather than average response goals).

- **Performance goal importance**. The Workload Manager will attempt to allocate system resources so that the most important performance goals are met. In order to accomplish this, the performance goals must be adequately partitioned into highest importance down to lowest importance. Additionally, most organizations have workload which can be classified as discretionary.

  If too many service classes (and too much workload) are assigned high performance goals, the Workload Manager will be unable to meet all performance goals.

- **System resources**. The Workload Manager will attempt to allocate resources to meet performance goals. However, the Workload Manager cannot allocate resources if insufficient resources exist to meet the requirements of the workload. You might have to acquire additional resources, reschedule workload, or shift workload to other systems in order for the Workload Manager to satisfy the performance goals you have defined.

  This problem is listed last because CPExpert believes that you should make all possible tuning efforts before acquiring additional system resources.

Reference: MVS/ESA SP Version 5 Planning: Workload Management

MVS/ESA SP Version 5 Programming: Workload Management Services

OS/390 MVS Planning: Workload Management

OS/390 MVS Programming: Workload Management Services

z/OS MVS Planning: Workload Management

z/OS MVS Programming: Workload Management Services

Rule WLM002: Conflict exists between Service Class and Report Class

Finding: CPExpert has detected that the service policy being analyzed assigns a workload to a **Report Class**. However, the service policy member being analyzed contains a **Service Class** with the same name.

Impact: This finding should be viewed as NO IMPACT on the performance of your computer system. However, the workload information recorded by SMF may contain "double counting" of data. Consequently, the finding could have a HIGH IMPACT on accounting for the use of system resources, on billing for use of system resources, or on capacity planning efforts.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Two types of performance classes are reflected in data maintained by the Workload Manager: *service classes* and *report classes*.

- Service classes are used by the Workload Manager to determine whether performance goals are being met and to control the allocation of system resources to address spaces[1]. A single transaction, job, Started Task, etc. can be controlled by only one service class (defined in the service policy).

- Report classes can be used to obtain information about the use of system resources at various levels of detail, based upon the specifications contained in the Workload Group definition of the service policy. Up to 999 report classes may be defined in a service policy. Earlier versions of MVS allowed the assignment of a workload element to up to four report performance groups. With MVS/ESA SP5, a single transaction, job, Started Task, etc. can be assigned to only **one** report class.

The Workload Manager ISPF application does not verify that a report class has a unique name; a report class and a service class may be defined with the same name. Additionally, the Workload Manager does not verify that the report class and service class names describe the same workload. A

[1]The Workload Manager may further classify service classes into "served" service classes and "server" service classes. CICS or IMS transactions may be placed into unique service classes, if CICS Version 4 or IMS Version 5 are installed. The Workload Manager will monitor the performance of the transactions against the goal and goal importance associated with their service class. However, the Workload Manager cannot assign resources to the "served" transactions, as the transactions do not directly represent address spaces. CICS or IMS regions act as "servers" for the transactions. If the "served" transactions are not meeting performance goals, the Workload Manager may assign resources to the "server" service class consisting of the CICS or IMS region(s). Please see Section 4 for additional discussion.

©Copyright 1994, Computer Management Sciences, Inc.

report class could have the same name as a service class, but the two classes could describe completely different workloads.

Double counting of resource usage in SMF post-processing can occur if a transaction is assigned to a report class and the same name for the report class is assigned to a service class. Double counting of resource usage could have a significant impact on overall accounting for the use of system resources, on billing for use of system resources, or on capacity planning efforts.

Additionally, analysts or managers can become confused when reviewing reports if the same name were to be used in describing different workloads.

The service policy being analyzed by CPExpert contains a report class definition which has the same name as a service class. CPExpert produces Rule WLM002 to alert you to this conflict.

The following example illustrates the output from Rule WLM002:

```
RULE WLM002:  CONFLICT EXISTS BETWEEN SERVICE CLASS AND REPORT CLASS

   The Service Policy being analyzed (Policy EQUALHRD) contains a Report
   Class which conflicts with a Service Class.  This conflict could cause
   double accounting of system resources in reports which key off of the
   class name.  This finding has no known effect on performance, but could
   have a HIGH IMPACT on accounting, billing, or capacity planning.  The
   following Report Class conflicts with a Service Class of the same name:

            REPORT CLASS
               IMS
               IMS1
               TSO
```

Suggestion: CPExpert strongly suggests that you eliminate the conflict between the report class and service class in the service definition.

Reference: MVS Planning: Workload Management

OS/390 (V2R6):	Chapter 10: Defining Report Classes
OS/390 (V2R7):	Chapter 10: Defining Report Classes
OS/390 (V2R8):	Chapter 10: Defining Report Classes
OS/390 (V2R9):	Chapter 10: Defining Report Classes
OS/390 (V2R10):	Chapter 10: Defining Report Classes
z/OS (V1R1):	Chapter 10: Defining Report Classes
z/OS (V1R2):	Chapter 10: Defining Report Classes
z/OS (V1R3):	Chapter 10: Defining Report Classes
z/OS (V1R4):	Chapter 10: Defining Report Classes

Rule WLM003: The service policy was changed

Finding: CPExpert has detected that the service policy was changed at the time shown by this rule.

Impact: This finding should be viewed a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact will depend upon (1) the number of changes and (2) whether service classes were altered in the new service policy.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Users communicate their service requirements for workloads by defining a *Service Policy* to the Workload Manager. A service policy is simply a "named" collection of service classes and their associated performance goals and (optionally) processing boundaries for the address spaces associated with the service classes.

A service policy is created using the Workload Manger ISPF panel. Once the service policy is created, it is stored in the couple data set. The policy can then be activated by the Workload Manager ISPF application *Policy Selection List* panel. The policy can also be activated by an operator issuing the VARY command and specifying the name of the new policy (for example, V WLM,POLICY=newpolicy).

A policy applies to a sysplex. Once a policy is activated, the definitions contained in the policy remain in effect for the duration of the IPL. Although part or all of the policy may be overridden by a different policy, any definition not overridden will remain in effect.

There are three performance-related effects inherent in a policy change:

- The Workload Manager and System Resources Manager must begin collecting new statistics to describe the new service policy. Consequently, a relatively long time may be required for the WLM and SRM to again make decisions about managing system resources.

- Any service class affected by the policy change will be "reinitialized" with respect to the service accumulated and all address spaces will in the service class will automatically be reassigned to Period 1 (if multiple periods exist). This effect may harm performance in an interactive environment. Address spaces which accumulate large amounts of system resources may conflict with interactive address spaces until the

©Copyright 1994, Computer Management Sciences, Inc.

address spaces consuming large amounts of system resources accumulate sufficient service units to again move to lower service class periods.

- RMF creates interval records (SMF Type 72 records) to record the information accumulated under the old service policy. RMF will again write interval records at the normal end of the RMF measurement interval. Thus, two sets of interval records will be written by RMF when a service policy change occurs: one set covering the interval from the last normal RMF recording interval up to the service policy change and one set covering the interval from the new service policy activation until the next normal RMF recording interval. Data analysis (by either CPExpert or an analyst) during these short periods must be viewed with caution.

The following example illustrates the output from Rule WLM003:

```
RULE WLM003:  THE SERVICE POLICY WAS CHANGED

   CPExpert has detected that a service policy change occurred in the below
   measurement intervals.  When the service policy is changed, RMF creates
   interval records for data collected up to that point.  RMF then will
   create another set of interval records when the normal measurement
   interval ends.  Thus, two sets of RMF data were created during the below
   measurement interval.  Data analysis for this period must be viewed with
   caution, as some of the calculations may be distorted by the short
   measurement intervals.

   EFFECTIVE TIME OF CHANGE      OLD SERVICE POLICY     NEW SERVICE POLICY
    07DEC1994:11:29:05                HPTSPOL1               HPTSPOL1
```

In the above example, the service policy **name** did not change. However, CPExpert detected that the service policy activation date/time (SMF Type 72 variable R723MTPA) had changed, indicating that the service policy had been activated at the indicated time. The active service policy can be copied to a temporary data set, the policy in the temporary data set can be changed, the changed policy installed in the couple data set, and the policy then activated. Thus, a change in the policy activation date/time signals that the policy itself has changed, regardless of whether the name changed.

Suggestion: There are no suggestions associated with this finding. CPExpert produces the information to alert you to be cautious about the analysis performed.

Reference: MVS Planning: Workload Management
MVS/ESA(SP 5): Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

Rule WLM004: CPExpert believes too many service policy changes occurred

Finding: CPExpert has detected that the service policy was changed relatively frequently.

Impact: This finding should be viewed a MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact will depend upon (1) the number of changes and (2) whether service classes were altered in the new service policy.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM003: The service policy was changed

Discussion: Please refer to Rule WLM003 for a discussion of service policy changes and the harmful effect on system performance of the changes.

CPExpert accumulates the number of service policy changes. CPExpert produces Rule WLM004 if the number of changes per day is greater than the **POLCHG** guidance variable in USOURCE(WLMGUIDE). The default value for the POLCHG guidance variable is 3, indicating that no more than 3 policy changes should occur during any interval being analyzed by CPExpert, or 3 changes per day.

The following example illustrates the output from Rule WLM004:

```
RULE WLM004:  CPEXPERT BELIEVES TOO MANY SERVICE POLICY CHANGES OCCURRED

   CPExpert has detected that the service policy was changed 6 times
   during the interval being analyzed.  The service policy should be
   infrequently changed.  Policy changes cause the Workload Manager to be
   less effective in managing system resources to meet performance goals
   (goals may change, historical data must be discarded and new data must
   be collected before decisions can be made, etc.).  Additionally, RMF
   data become less reliable because irregular measurement intervals are
   created when service policy changes occur.  CPExpert suggests that the
   number of service policy changes be reduced.
```

Suggestion: Service policy changes should occur relatively infrequently, as any service policy change causes some adverse effect on performance simply because the policy changed. A relatively large number of changes normally should be cause for management action to reduce the changes.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert realizes that there may unique situations in which management policy dictates that the service policy should change more often than the guidance variable. You can change the guidance to CPExpert if you have such an unusual situation. The **POLCHG** guidance variable in USOURCE(WLMGUIDE) can be used to provide guidance to CPExpert regarding the number of service policy changes you consider acceptable.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 5: Defining Service Policies
OS/390 (V1R1):	Chapter 5: Defining Service Policies
OS/390 (V1R2):	Chapter 5: Defining Service Policies
OS/390 (V1R3):	Chapter 5: Defining Service Policies
OS/390 (V2R4):	Chapter 5: Defining Service Policies
OS/390 (V2R5):	Chapter 5: Defining Service Policies
OS/390 (V2R6):	Chapter 5: Defining Service Policies
OS/390 (V2R7):	Chapter 5: Defining Service Policies
OS/390 (V2R8):	Chapter 5: Defining Service Policies
OS/390 (V2R9):	Chapter 5: Defining Service Policies
OS/390 (V2R10):	Chapter 5: Defining Service Policies
z/OS (V1R1):	Chapter 5: Defining Service Policies
z/OS (V1R2):	Chapter 5: Defining Service Policies
z/OS (V1R3):	Chapter 5: Defining Service Policies
z/OS (V1R4):	Chapter 5: Defining Service Policies

Rule WLM005: Velocity Goal may be too high for batch service class

Finding: CPExpert noticed that a high velocity goal was specified for a service class, and CPExpert believes that the service class describes batch workload.

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: Installations may specify an *execution velocity goal* for a service class. An execution velocity goal is a measure of how fast work should run when the work is **ready** to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage[1]. In a sense, the execution velocity combines many controls available in pre-Goal Mode MVS environments into a single metric.

The purpose of specifying an execution velocity is to allow installations to specify how important it is to have work being processed, when the work has no time-related measure (that is, a response requirement is not associated with the work). An execution velocity goal is suitable for most batch work and most started tasks.

Many installations will specify an execution velocity goal for batch work, and base the actual goal on the batch execution velocity presented in the RMF *Workload Activity Report* produced in Compatibility Mode. The batch execution velocity in the RMF *Workload Activity Report* reflects what **was achieved** by the system executing in Compatibility Mode, but does not necessarily reflect **what should be achieved**.

There often is a dramatic difference between what **was** achieved and what **should be** achieved. Batch work often will achieve a relatively high execution velocity. This execution velocity is not often suitable for a batch performance goal unless the workload is extremely stable.

For example, the resource requirements of system workload often vary from day to day or week to week. This variance may allow batch work executing in Compatibility Mode to achieve a high execution velocity in periods of relatively low overall system demand. This high execution velocity may not be required to meet installation objectives.

[1]Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

©Copyright 1994, Computer Management Sciences, Inc.

A high execution velocity goal can cause significant system problems if there is the possibility of batch work (or other types of work, for that matter) to be erratic in nature. For example, if batch jobs can enter into a CPU loop, the batch workload may "seize" the system for whatever percentage was specified as the execution velocity. To illustrate, suppose that an execution velocity of 50 was specified for a batch service class. A CPU-intensive batch job (or CPU-looping job) could require 50% of the CPU and deny CPU access to all work of equal or lower importance.

IBM Workload Manager developers have suggested that an execution velocity of 10 or 20 should be adequate for most batch service classes. These values are based on their observations that most batch jobs are naturally I/O intensive, rather than CPU intensive.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "batch" and assumes that the service class describes batch workload if "batch" is encountered. Two considerations apply:

- It is, of course, possible that the service class does not describe batch workload even though "batch" is in the description. This instance is unlikely, as most installations will use the word "batch" in a description of only batch work.

- It also is possible that the word "batch" will not be in the description of a service class of batch workload. CPExpert will be unable to detect these situations.

If CPExpert detects "batch" in the Service Class Description, CPExpert examines the performance goal type to see whether the goal type is execution velocity. If the goal type is execution velocity, CPExpert examines the performance goal. CPExpert produces Rule WLM005 if the performance goal is greater than the **MAXVEL** guidance variable in USOURCE(WLMGUIDE).

The default value for the MAXVEL guidance variable is 20, indicating that Rule WLM005 will be produced if more than 20 had been specified as an execution velocity goal for a service class containing batch workload.

The following example illustrates the output from Rule WLM005:

```
RULE WLM005:  VELOCITY GOAL MAY BE TOO HIGH FOR BATCH SERVICE CLASS

   CPExpert noticed that BATCHA Service Class (Period 1) had an execution
   velocity goal of 30.  The BATCHA Service Class had the word "batch"
   in its Service Class Description.  Consequently CPExpert assumes that
   the service class consists of batch jobs.  Specifying a relatively
   high velocity goal of 30 for batch jobs may cause performance problems
   unless the batch jobs are well-behaved.  Under some circumstances, this
   velocity goal could result in 30% of the system being used by batch
   workload.  Please refer to Rule WLM005 in the WLM Component User Manual
   for a discussion of this issue.
```

Suggestion: CPExpert suggests that you consider lowering the execution velocity goal for the service class containing batch workload.

Alternatively, if you believe that your specification is appropriate, you can use the MAXVEL guidance variable to provide guidance to CPExpert. Since 99 is the maximum execution velocity specification for the Workload Manager, you can disable Rule WLM005 by specifying a value of 99 for the MAXVEL guidance variable.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"MVS WLM: What it is and what you should know to use it", Bernie Pierce (Candle Corporation; previously at IBM Corporation as Workload Manager Developer), 1995 SHARE Winter Meeting

Rule WLM006: The response performance goal may be too large

Finding: CPExpert believes that the response performance goal specified for a service class may be too large.

Impact: This finding should be viewed a LOW IMPACT or MEDIUM IMPACT on the performance of your computer system. The finding could have a HIGH impact on the performance of the service class identified by this finding.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Users specify a performance goal for each service class. There are four types of performance goals: average response, percentile response, execution velocity, and discretionary. The first two (average response and percentile response) are the subject of this rule description.

The Workload Manager ISPF Response Time Goal Panel allows a response performance goal of up to 24 hours to be specified. Response goals in minutes or hours are typically associated with batch workloads.

CPExpert believes that a response performance goal of over 5 minutes is likely to result in unsatisfactory performance in most environments and a response goal of **less than 1 minute** is more likely to yield desired results. The following discussion explains why CPExpert believes that relatively long response goals are inappropriate:

- The Workload Manager attempts to adjust system resources as necessary to achieve the performance goal specified for service classes. The Workload Manager evaluates how well the existing resource policy allows performance goals to be met **every 10 seconds**. This 10-second process is called the *Policy Adjustment Interval*.

  During policy adjustment, the Workload Manager evaluates the performance of each service class. The evaluation is accomplished by computing a Performance Index for each service class period and analyzing the Performance Index within each level of Goal Importance[1]

  Obviously, in order to analyze how well a service class is performing against a response goal, one or more "transactions" must have completed during the previous interval. If no transactions completed, the Workload Manager has no information on which to assess the

[1]Please refer to Section 4 for a more comprehensive discussion of the Policy Adjustment process.

©Copyright 1994, Computer Management Sciences, Inc.

performance of the service class with respect to the response goal. In fact, the process works much better if *many* transactions complete, as the Workload Manager can compute average response or percentile response based on a larger sample of work.

Once the Workload Manager makes a policy adjustment decision, it evaluates the effect of that decision during the **next** policy adjustment interval. In order to assess the effect of the decision on response time, multiple transactions must complete so the Workload Manager can evaluate the effect on transaction response[2] time.

In summary, the Workload Manager must detect that a response goal was missed, and take action to improve performance for the service class missing its goal. Then the WLM must determine whether the action helped, or whether additional actions must be taken. This cycle can continue for awhile. With short response goals and lots of transactions, the WLM will have adequate performance data (many ended transactions yielding response information) to evaluate, and will have quick feedback on how well its decisions helped the service class meet its responses goal. The WLM can detect/adjust/evaluate/adjust relatively frequently with respect to the goal. Still, the detect/adjust/evaluate happens only once per 10 seconds.

This process works extremely well if the transactions represent interactive work (e.g., TSO transactions, CICS transactions, or IMS transactions). Many transactions normally will complete in a policy adjustment interval and the Workload Manager will have adequate information upon which to assess the results of the policy decisions.

If the "transaction" really is a batch job with a relatively long response performance goal, it is unlikely that many transactions will complete in the 10-second policy adjustment interval. Thus, the Workload Manager has little or no information on which to base its policy adjustment decisions; the Workload Manager must wait for batch jobs to complete before any decisions can be made. Consequently, the Workload Manager will be unresponsive in adjusting system resources to meet the performance goal for the batch jobs.

- Consider that the WLM makes resource adjustment decisions every 10 seconds. These adjustments are partially based on how well work meets goals (other factors are general housekeeping, etc.).

[2]The Workload Manager can assess the effect of some policy decisions without response-related information. For example, suppose that the Workload Manager determined that paging was a major cause of performance degradation. The Workload Manager might make processor storage decisions to either protect or restrict central or expanded storage for certain service classes. The effect of these decisions would be apparent from a system view (e.g., paging increased or decreased) without requiring transaction response data. However, the Workload Manager cannot determine whether the overall response performance goal is being met until transactions complete.

Suppose that a few batch jobs execute in a service class period with a *response goal of 20 minutes*. It takes awhile[3] for the WLM to detect that a goal was missed. If, for example, a job took over an hour to complete, it would take more than an hour for the WLM to recognize that work in the service class period was missing its response goal.

Recall that the WLM is going to make decisions every 10 seconds, and the WLM then must determine whether the decisions helped improve response time. In this example, more than 20 minutes additional must lapse before the WLM can have data to figure out whether its decisions were appropriate! In fact, at least one transaction must end before the WLM can assess whether the performance goal had been achieved. If a new job would take an hour to complete (meaning that the policy adjustment decisions did not help), it might take the WLM that hour to determine that its policy adjustment was not effective.

It is true that the WLM can make resource allocation decisions based on observed delays to the long-running work (denied CPU use, paging delays, I/O delays, etc.), and the WLM can dynamically assess whether the work was being delayed less because of decisions related to these CPU delays, paging delays, I/O delays, etc. Consequently, the WLM can "guess" that performance is improving based on decreased delay to the work. However, that is exactly what execution velocity takes into account. This means that for long transaction response times, the WLM in effect implements velocity goal management.

This “implicit” implementing velocity goal management is not as effective as explicitly stating a velocity goal. This is because it takes too long (the duration of the response goal) for the WLM to detect that a response goal was missed, whereas execution velocity goals would be computed every 10 seconds.

- The Workload Manager evaluates system performance considering the performance of all service classes, based on their level of importance. Most modern computer environments have a mix of workload, consisting of both interactive and non-interactive. The interactive workload usually has a higher importance, and interactive workload often is quite dynamic in terms of system requirements.

  One consequence of this nature of interactive work is that the Workload Manager typically will adjust resource allocation policies based on the requirements of the interactive workload. Only in the most stable environments will policy adjustment decisions be driven by relatively lengthy non-interactive response goals.

[3] That time would be more than 20 minutes, since at least one batch job must complete and exceed its 20 minutes goal before the WLM could detect that the goal had been missed.

- Workload Manager developers have stated that only 20 minutes of historical information are retained by the Workload Manager. At present, it is unclear how the internal Workload Manager algorithms discard data and it is unclear what effect discarding data has on lengthy response goals.

  Very short transactions are typically homogeneous with respect to their execution characteristics so the WLM does not have to worry about radically differing use of processor or I/O amongst the different transactions. Even if some transactions use radically differing resources from the general population, their effect will be minuscule because they end so quickly. Consequently, an adjustment decision can be made without worry that the resource demands will radically change from one "transaction" to the next.

  These characteristics of short transactions do not normally apply with batch work or other work that consists of long-running transactions. Any particular long-running batch job is not necessarily homogenous with the batch job population, with respect to its use of system resources. Also, unlike short transactions, long-running work does not tend to be homogeneous, and there often is drastic differences in the resource requirements among long-running jobs.

  This heterogeneous nature of long-running work would often result in the WLM making policy adjustment decisions, based on resource consumption characteristics of **ended** long-running work. However, the **currently-running** work might not have similar resource demands and delays that the WLM had observed from the ended work.

- Please note that IBM's *MVS/ESA SP Version 5: Planning: Workload Management* specifically states "Work that is appropriate for a response goal should have a reasonable number of transaction completions over 20 minutes of time. If there are only a few completions, you are better off using a velocity goal."

- You might have defined service classes to describe subsystem transactions (such as CICS transactions) which have long Idle state times. Rule WLM122 describes transactions with long Idle state times, and suggests an approach which includes defining a **very long** response goal for the service class containing these transactions. CPExpert suppresses Rule WLM006 for transaction subsystem service classes.

The following example illustrates the output from Rule WLM006:

```
RULE WLM006:  THE RESPONSE PERFORMANCE GOAL MAY BE TOO LARGE

 CPExpert noticed that BATDEVHI Service Class (Period 1) had a response
 goal of 0:20:00.0.  This response goal is quite large relative to the
 intervals in which the Workload Manager makes system adjustments.  It
 is unlikely that the Workload Manager's actions will be effective with
 such a large response goal.  You may have better success by specifying
 an execution velocity goal for this service class.  Please refer to
 Rule WLM006 in the WLM Component User Manual for a discussion of this
 issue.  Rule WLM173 in this output will show any measurement intervals
 when CPExpert believed that the large response goal could be a cause
 of the service class missing its response goal.  If Rule WLM173 is not
 produced, the large response goal probably had no adverse effect on
 this service class period.
```

Suggestion: CPExpert suggests that you consider the following alternatives if Rule WLM006 is produced:

- Specify an execution velocity goal for the service class identified by Rule WLM006. This normally is the best alternative. There are exceptions to this general advice, as discussed below.

- Specifying ANY goal automatically means that work elements in the service class will be assigned to the range of dispatching priorities reserved for "goal" work. This means that the work will always have a higher dispatching priority than discretionary work. Consequently, specifying a long response goal could be a valid approach if you want to always make sure that the work has a higher dispatching priority than discretionary. As described earlier, however, this is not normally a good solution since specifying an execution velocity goal (even a small velocity) would provide better WLM actions.

- Specifying a long response goal causes the work to be a candidate for Discretionary Management. While the work will always have a higher dispatching *priority* than discretionary, a very high goal **could** cause it to have a Performance Index less than 0.7 (which is the Performance Index when cap slice capping can start), and stops internal resource capping when the Performance Index is greater than or equal to 0.81 (which is the internally-used Performance Index for discretionary work). Consequently, the work with a long response goal can be subject to the Discretionary Management Cap Slice algorithm, to allow discretionary to periodically have access to the CPU. Again, this is not normally a good solution since specifying an execution velocity goal (even a small velocity) would provide better WLM actions.

- **Rule WLM173** will show any measurement intervals when CPExpert believed that the large response goal could be a cause of the service class missing its response goal. If Rule WLM173 is not produced, the

large response goal probably had no adverse effect on this service class period.

- You can adjust (or turn off) this analysis if you disagree with CPExpert's reasoning. The **MAXRESP** guidance variable in USOURCE(WLMGUIDE) can be used to provide guidance to CPExpert on the maximum response performance goal which CPExpert views as acceptable.

  The default specification for the MAXRESP guidance variable in WLMGUIDE is %LET MAXRESP=0:05:00, indicating that CPExpert that any response performance goal greater than 5 minutes causes Rule WLM006 to be produced. You could “turn off” this rule by specifying %LET MAXRESP=24:00:00 in USOURCE(GENGUIDE). Since a response goal cannot be larger than 24 hours, this would have the effect of “turning off’ CPExpert’s analysis in this area.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

Rule WLM007: MSO Service Coefficient may be too large

Finding: CPExpert has determined that the MSO service coefficient in the service definition may be too large.

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of your computer system. However, under certain circumstances (explained below), this finding could have a HIGH IMPACT on the performance of specific application systems, and may have a HIGH IMPACT on overall system performance.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The MSO service coefficient specification in the service definition indicates the value by which accumulated storage service units will be multiplied (or weighted) before the storage service units are included in the accumulated service associated with an address space.

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

service = (**CPU** x CPU Service Units)
+ (**IOC** x I/O Service Units)
+ (**MSO** x Storage Service Units)
+ (**SRB** X SRB Service Units)

The CPU, IOC, MSO, and SRB values highlighted in the above formula relate to the corresponding values specified for the CPU, IOC, MSO, and SRB service coefficients specified in the Workload Manager ISPF panel. The default values for these coefficients are CPU=10, IOC=5, MSO=0, and SRB=10.

The basic intent of the service consumption algorithm is to create a single value which corresponds to the rate at which an address space consumes system resources. As this rate increases, an installation can chose to assign the address space a lower priority with respect to other address spaces. One result of this approach is that the service consumed by address spaces can be controlled: important work can be given access to system resources and the access of less important work can be restricted. This control is accomplished between domains and within domains.

©Copyright 1994, Computer Management Sciences, Inc.

With MVS/ESA SP5.1, the primary use of the accumulated service associated with an address space is to determine whether the address space is moved to a different service class period.

When the address space accumulates more than a specified amount of service, the SRM will reassign the address space to the next lower performance period (if multiple periods are defined)[1]. The amount of service controlling when an address space is reassigned to the next lower performance period is specified by the value of the **DUR** keyword in the definition of the service class performance period.

The normal purpose of defining multiple performance periods is to give higher priority (smaller response goals or higher importance) to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources.

Of particular importance to this finding is the way in which Storage Service Units are computed. The Storage Service Units are computed as:

$$Storage\ Service\ Units = \frac{(real\ page\ frames)\ x\ (CPU\ service\ units)}{50}$$

The intent of dividing by 50 was to cause the Storage Service Units value to be scaled comparable to the other service units. This intent worked fairly well for systems with a relatively small amount of central storage.

However, the Storage Service Unit values can become quite large and can be unpredictable in systems with a significant amount of central storage. These two potential effects are discussed separately:

- The Storage Service Unit values can become large in systems with a significant amount of central storage. An address space can be allocated a large amount of central storage, so that the Storage Service Unit values completely dominate the Service Unit computation.

[1]Note that the address space is not reassigned immediately upon accumulation of the specified amount of service. The reassignment is performed only when the SRM evaluates the address space for changed conditions. Therefore, the SRM cannot determine when the **exact** amount of service specified in the DUR parameter is used; the SRM can only determine when the DUR value is exceeded.

The SRM checks for transactions exceeding their DUR value only when the SRM gains control. Each address space normally is evaluated every three SRM seconds, and reassignment is performed only if the address space has accumulated more than the value specified for the DUR keyword. Consequently, an address space will always accumulate **more** service than specified by the DUR keyword before being reassigned.

©Copyright 1994, Computer Management Sciences, Inc.

For example, an address space may require an allocation of 500 page frames of central storage. Suppose that the address space used 10 milliseconds of CPU (total TCB and SRB) time per second and issued 10 I/O operations per second. If the default service coefficients were used and the processor was a 3090(Model 600J single image), the service rate would be calculated as:

$$service = (10 * 0.01 * 941.53) + (5 * 10) + \frac{(3 * 500 * 0.01 * 941.53)}{50}$$

$$service = 94.153 + 50 + 282.459$$

$$service = 426.612$$

As the above calculation illustrates, the Storage Service Unit portion accounts for about three times as much of the service as does the CPU Service Unit portion in the example.

The *Workload Manager: Planning* document notes that a large MSO value can cause SRM variables to overflow. In fact, the SRM monitors certain variables to determine whether they are in danger of overflowing. A long-running or never-ending "transaction" is stopped and restarted if the SRM determines that variables are in danger of overflowing. A large MSO service coefficient is the primary cause of variables overflowing.

- The Storage Service Unit values can be unpredictable in systems with significant central storage. This situation can arise because of varying demands for central storage.

  - Suppose that on one occasion, a particular address space executes in a workload mix in which central storage is heavily used. The central storage allocated to the address space would regularly be restricted to the basic working set of the address space. During this execution, the address space would have a service rate which might correspond to the original intent of the service rate computation.

  - However, suppose that on another occasion, the address space executes in a workload mix in which central storage is not heavily used. The central storage allocated to the address space might include many relatively "old" page frames - rather than simply the basic working set of the address space. During this execution, the address space would have a service rate that could be many times

the service rate in the hypothetical situation in the preceding paragraph.

Under some circumstances, this situation would not be harmful. However, the SRM might place the address space into a lower service class period, with the performance goals and importance associated with the new service class period.

If the service rate is unpredictable and is unrelated to the actual requirements of the address space, then the address space will have unpredictable access to system resources. This unpredictable access is contrary to the basic intent of the management philosophy behind the design of the SRM and of the Workload Manager.

The following example illustrates the output from Rule WLM007:

```
RULE WLM007:  THE MSO SERVICE COEFFICIENT MAY BE TOO LARGE

   The MSO service coefficient was specified as 3, and CPExpert believes
   that this value is too large.  The MSO service coefficient is used to
   adjust the storage service portion of the overall service attributed by
   address spaces and to service classes.  The algorithm used by the SRM
   was designed when central storage was a major constraint in computer
   systems.  With increasingly large amounts of processor storage, central
   storage is not usually a major constraint to system performance.
   Consequently, installations often should reduce the  contribution that
   storage service makes to the overall computed service associated with
   address spaces and service classes.  Further, a large MSO coefficient
   used in modern systems can cause SMF or SRM fields to overflow, with
   resulting loss of data or erroneous actions by the Workload Manager or
   the SRM.
```

Suggestion: CPExpert suggests that you verify the specification for the MSO service coefficient in the service definition being analyzed.

If you are executing with a large central storage, you may wish to reduce the MSO service coefficient to zero. Reducing the MSO service coefficient will reduce the effect of central storage page frames on the service rate associated with an address space.

If you do change the MSO service coefficient, please review the DUR values you have specified for any multiple-period performance groups to make sure that the DUR values are appropriate with the decreased emphasis on storage service.

Reference: MVS Planning: Workload Management
MVS/ESA(SP 5): Chapter 12: Defining Service Units and Coefficients

OS/390 (V1R1):	Chapter 12: Defining Service Units and Coefficients
OS/390 (V1R2):	Chapter 12: Defining Service Units and Coefficients
OS/390 (V1R3):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R4):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R5):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R6):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R7):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R8):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R9):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R10):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R1):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R2):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R3):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R4):	Chapter 11: Defining Service Units and Coefficients

Rule WLM008: DUR value may be too large for TSO Period 1

Finding: CPExpert believes that the DUR value may be too large for a service class that CPExpert believes describes TSO Period 1 transactions.

Impact: This finding has a MEDIUM impact or HIGH impact on the performance of TSO interactive transactions.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The Workload Manager allows installations to group similar work into service classes. For example, TSO transactions might be placed into one service class, batch jobs might be placed into another service class, etc. Users define service classes by specifying the name of the service class and specifying the performance goal, service class, and service class importance.

An installation may wish to assign different performance attributes to work depending upon the inherent resource demands of the work. For example, all TSO transactions might be assigned to a particular service class. However, installation management might wish to assign a high performance goal and goal importance to trivial TSO transactions, non-trivial (but still quite interactive) TSO transactions might be assigned a slightly lower performance goal and goal importance, and relatively non-interactive TSO transactions might be assigned still lower performance goal and goal importance. This differentiation among TSO transactions **based on the inherent characteristics of the transactions** is accomplished via the "service class period" option.

A service class may be broken into multiple service class **periods**. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional service class periods (although typically no more than two or three additional service class periods are defined). Each service class period can have its own performance goals, defined to the Workload Manager via the Workload Manager ISPF panels.

An address space (TSO transaction, batch job step, etc.) begins in Period 1 of the service class to which it is assigned. The address space transitions from Period 1 to Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction. The "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources. The actual resources used by the

address space are adjusted by the CPU, IOC, SRB, and MSO service coefficients to yield the CPU, I/O, and memory "service" requirements of the address space.

When the address space accumulates more than a specified amount of service, the address space will be reassigned to the next lower service class period (if multiple periods are defined)[1]. The amount of service controlling when an address space is reassigned to the next lower service class period is specified by the value of the **DUR** keyword in the service definition. The DUR keyword is associated with any service class period which has subsequent service class periods.

The normal purpose of defining multiple service class periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources.

It is instructive to examine the resources that may be in conflict. The resources mainly consist of CPU cycles, processor storage, and I/O operations.

- Prior to MVS/ESA SP5, the CPU resource was the main area of conflict. CPU dispatching priority is established at the performance group period level. An address space in a particular performance group period executed at the same CPU dispatching priority as all other address spaces in the performance group period[2]. Non-interactive transactions could cause interactive transactions to be denied access to a processor for an unacceptable interval.

- Conflict with processor storage occurs when non-interactive transactions require either central storage or expanded storage, causing central storage page fault or expanded storage delays for interactive transactions. Non-interactive transactions often (but not always) require a significant amount of processor storage.

---

[1]Note that the address space is not reassigned immediately upon accumulation of the specified amount of service. The reassignment is performed only when the System Resources Manager (SRM) evaluates the address space for changed conditions. Therefore, the SRM cannot determine when the **exact** amount of service specified in the DUR parameter is used; the SRM can only determine when the DUR value is exceeded.

The SRM checks for transactions exceeding their DUR value only when the SRM gains control. Each address space normally is evaluated every three SRM seconds, and reassignment is performed only if the address space has accumulated more than the value specified for the DUR keyword. Consequently, an address space will always accumulate **more** service than specified by the DUR keyword before being reassigned.

[2]This discussion ignores the implications of time-slicing algorithms and ignores the implications of the Mean-time-to-Wait (MTTW) dispatching algorithms. These special case algorithms are not normally used for Period 1 or Period 2 of multi-period performance groups.

- Non-interactive transactions can cause conflict with I/O operations by either path, controller, or device delays to interactive transactions. Fortunately, (1) modern systems normally have multiple paths to devices and path delay is not a serious concern, (2) and non-interactive transactions normally do not access the same devices as interactive transactions, and (3) interactive transactions normally do not execute many I/O operations. The combined effect is that conflict for I/O operations rarely is a cause of performance problems between interactive and non-interactive transactions.

As the above discussion illustrates, the main concern prior to MVS/ESA SP5 was CPU dispatching. This concern has been almost completely eliminated with SP5.

- Prior to SP5, CPU dispatching was on a "first come first served" or FIFO basis from the Dispatcher's True Ready Queue. A non-interactive transaction could easily deny an interactive transaction from access to a processor for a prolonged period; the non-interactive transaction could retain control of a processor until it either completed or it voluntarily entered a Wait state.

- With SP5, the Dispatcher algorithm has been redesigned. One aspect of the redesign is that each interrupted dispatchable unit (a TCB or SRB) is placed at the end of the dispatching queue for its current dispatching priority.

  For example, suppose that a dispatchable unit was executing with a priority of 240. When the dispatchable unit was interrupted, it would be placed behind all other dispatchable units on the dispatch queue with a dispatching priority of 240 (but ahead of all dispatchable units with a dispatching priority of 239 and lower).

  Since interrupts occur frequently, there is a constant adjustment of dispatching order. This readjustment requires practically no additional overhead, as only queue pointers are being manipulated.

  One effect of the redesign is to prevent any dispatchable unit from "seizing" a processor and denying access to other dispatchable units at the same priority. Since a dispatchable unit cannot deny access to other dispatchable units, a non-interactive transaction cannot deny CPU access to a Ready interactive transaction.

Consequently, with SP5, there is less need to "separate" non-interactive and interactive transactions, and there is less need to specify multiple periods. Nonetheless, the need to separate interactive and non-interactive transactions still exists.

The SRM will be able to differentiate between interactive and non-interactive transactions only if the values specified for the DUR keyword roughly correspond to the resource requirements of trivial, interactive, and non-interactive transactions.

If the value specified for the DUR keyword for Period 1 is too large, non-trivial and non-interactive transactions will execute with the same performance management controls as those given to trivial transactions. This would defeat the purpose of breaking the service class into multiple service class periods.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "TSO" and assumes that the service class describes TSO transactions if "TSO" is encountered.

If CPExpert identifies a service class describing TSO transactions, CPExpert examines the DUR value and the service class period. If a DUR value is specified and the period is Period 1, CPExpert analyzes the DUR value[3]. CPExpert concludes that the DUR value is too large for TSO Period 1 transactions if the DUR value is greater than 100 (for service policies which have MSO less than 1) or greater than 300 (for service policies which have MSO 1 or higher).

This conclusion is based upon comparing the DUR value with the DUR values specified by other installations. After making this comparison, CPExpert found that the DUR value specified for TSO Period 1 was much higher than the value commonly specified for trivial TSO transactions.

The following example illustrates the output from Rule WLM008:

```
RULE WLM008:  THE DUR VALUE MAY BE TOO LARGE FOR TSO PERIOD 1.

   CPExpert noticed that TSOUSERS Service Class (Period 1) had the word
   "TSO" in its Service Class Description.  Consequently CPExpert assumes
   that the service class consists of TSO transactions.  The DUR value for
   the TSOUSERS Service Class was specified as 400.  CPExpert believes that
   the DUR value may be too large for TSO Period 1 transactions.  TSO
   Period 1 normally should handle interactive TSO transactions.  These
   interactive transactions typically use few system resources.  If the
   DUR value is too large for TSO Period 1, non-interactive transactions
   will remain in TSO Period 1 for a longer time before they are migrated
   to TSO Period 2.  During the time that the non-interactive transactions
   remain in TSO Period 1, they interfere with interactive transactions.
   If the TSOUSERS Service Class does describe TSO transactions, you may wish
   to reduce the DUR value so that Period 1 of this service class handles
   interactive TSO transactions and other periods handle less interactive
   TSO transactions.
```

[3]This analysis can be overridden by the **MAXDUR** guidance variable in USOURCE(WLMGUIDE).

Suggestion: CPExpert suggests that you review the DUR value specified for TSO Period 1. Unless there are unique conditions at your installation, CPExpert suggests that the DUR value be significantly lowered.

Most installations have established the DUR value for TSO Period 1 such that about 80-90% of the TSO transactions end in Period 1. There is nothing "magic" about these percentages! They are simply general guidance that has been provided by IBM. If your situation warrants a different percentage, set the DUR value accordingly.

The proper specification for the DUR value for TSO Period 1 depends entirely upon the resource requirements of trivial TSO transactions at your installation.

Some installations operate quite efficiently with only about 50% of the TSO transactions ending in Period 1! In fact, many installations have established an extremely low DUR value for TSO Period 1 (e.g., DUR=50) with the intent of rapidly processing extremely trivial TSO transactions (for example, a TSO user simply pressing a PF key in ISPF). More common specifications are DUR=100 for TSO Period 1.

The default analysis CPExpert makes depends upon the setting of the MSO service coefficient.

- If MSO is set to less than 1, CPExpert checks the DUR value against 100.

- If MSO is set to 1 or more, CPExpert checks the DUR against 300 as central storage use will represent a significant part of the service accumulation.

You can alter the analysis done by CPExpert by using the **MAXDUR** guidance variable in USOURCE(WLMGUIDE) to provide CPExpert with guidance about what DUR value you consider reasonable.

Reference: OS/390 (V1R3) MVS Planning: Workload Management
Chapter 8.2: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R4) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R5) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R6) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R7) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R8) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R9) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R10) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R1) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R2) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R3) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R4) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

Rule WLM009: Minimum CPU service assigned to Resource Group

Finding: CPExpert has determined that a minimum CPU service was specified for a resource group, and CPExpert believes that this minimum CPU service specification caused one or more service classes to miss their performance goal.

Impact: This finding should be viewed as generally having a HIGH IMPACT on the performance of your computer system.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM252: CPU access might be denied because of Resource Group minimum

Discussion: A resource group is simply a "named" description of the total minimum and maximum **unweighted** CPU service units per second which may be used by one or more service classes assigned to the resource group. A resource group is defined using the *Create a Resource Group* panel in the Workload Manager ISPF application. A resource group applies across an entire sysplex. Service class periods[1] are assigned to a resource group using the *Create a Service Class* panel in the Workload Manager ISPF application.

The Workload Manager will attempt to provide the specified minimum CPU service to the resource group. The Workload Manager attempts to provide the specified minimum CPU service to the resource group by adjusting the dispatching priority of service classes assigned to the resource group.

The Workload Manager will restrict service classes assigned to the resource group from using more than the specified maximum CPU service. The Workload Manager uses "CPU capping" to restrict the total amount of CPU time used by service classes assigned to the resource group.

There are potentially serious effects of specifying a minimum CPU service for a resource group. The effect is caused by the order in which the Workload Manager selects service classes for policy adjustment.

[1]A resource group may not be associated with a service class representing subsystem transactions (i.e., a service class defined for transactions executing under CICS/ESA Version 4.1 or under IMS/ESA Version 5). This is because CPU resources are not monitored by the SRM for the transactions; the CPU resources are monitored at the **address space level** (e.g., the CICS region or IMS message processing region). Further, CPU dispatching occurs at the address space level, rather than at the transaction level. Since CPU usage is not collected at the transaction level and CPU dispatching is at the address space level, the Workload Manager cannot control the amount of CPU resources allocated to service classes which represent transactions.

©Copyright 1994, Computer Management Sciences, Inc.

- The Workload Manager first determines whether any resource group is below the **minimum** CPU service specification. If the minimum CPU specification is not being provided, the Workload Manager takes the following actions in an attempt to provide the minimum CPU service:

  - The Workload Manager determines whether any service class assigned to the resource group is not meeting its performance goal. If any service class is not meeting its performance goal, the Workload Manager increases the dispatching priority (if appropriate) of the service class.

  - If no service classes assigned to the resource group were missing their performance goal, the Workload Manager increases the dispatching priority (if appropriate) of all service classes assigned to the resource group. The dispatching priority of all service classes assigned to the resource group (including those service classes with a discretionary goal[2]) may be increased.

- After the Workload Manager performs the above tasks, the Workload Manager may examine service classes based on the Goal Importance of the service classes.

The result of the above process can be that service classes with a low importance (or even service classes with a discretionary goal) can be assigned CPU dispatching priority above that which is assigned to the service classes with the highest Goal Importance! The resulting CPU dispatching priorities and CPU demands can result in service classes with high Goal Importance missing their performance goals.

This may not be the effect you wish, but the Workload Manager simply follows the specific direction provided for the resource group, namely, that a minimum CPU service was specified for the resource group and this minimum should be provided.

The discussion of Rule WLM252 describes how CPExpert concludes that the minimum CPU service specification for a resource group may cause a service class to miss its performance goal. The information provided with Rule WLM252 identifies the service class which missed its performance goal.

[2]Please note that the *MVS/ESA SP5 Planning: Workload Management* document is incorrect. This document states in the *Defining Resource Groups* section that "If there is a resource group defined for a service class with a discretionary goal, workload management achieves the minimum as long as the goals of work running in any other service class are not impacted. If other performance goals are impacted, then workload management does not maintain the minimum." Based on personal communication with the Workload Manager developer who wrote the specific code which attempts to provide the minimum specified CPU service, these statements are incorrect in the IBM document and the description provided above is what actually transpires.

Rule WLM252 also shows the amount of CPU time used by service classes at the same or lower Goal Importance as the service class missing its performance goal, and which were assigned to a resource group with a minimum CPU specification.

Please refer to the discussion in Rule WLM252 for additional information.

Rule WLM009 is provided to alert you to the possibility that there may be a problem with the service policy specifications. Rule WLM009 is produced only if Rule WLM252 was produced for any service class.

The following example illustrates the output from Rule WLM009:

```
RULE WLM009:  MINIMUM CPU SERVICE ASSIGNED TO RESOURCE GROUP.

   CPExpert believes that assigning a minimum CPU service to a Resource
   Group may have caused a service class to miss its performance goal.  At
   least one service class missed its performance goal and a significant
   reason for missing the performance goal was that the service class was
   denied access to a CPU.  At least one other service class was assigned
   to a Resource Group with a minimum CPU service specified, and the other
   service class was at the same importance or at a lower importance.  Rule
   WLM252 later in this report describes these situations.  You may wish to
   evaluate whether the Resource Group really should have a minimum CPU
   service specification, and whether the importance levels of the service
   classes are appropriate.
```

Suggestion: CPExpert suggests that you verify the minimum CPU service specification for resource groups defined in the service policy. Unless there are unique requirements for the minimum CPU service specifications, CPExpert suggests that the minimum be changed to zero.

Reference: MVS Planning: Workload Management

OS/390 (V2R4):	Chapter 7: Defining Resource Groups
OS/390 (V2R5):	Chapter 7: Defining Resource Groups
OS/390 (V2R6):	Chapter 7: Defining Resource Groups
OS/390 (V2R7):	Chapter 7: Defining Resource Groups
OS/390 (V2R8):	Chapter 7: Defining Resource Groups
OS/390 (V2R9):	Chapter 7: Defining Resource Groups
OS/390 (V2R10):	Chapter 7: Defining Resource Groups
z/OS (V1R1):	Chapter 7: Defining Resource Groups
z/OS (V1R2):	Chapter 7: Defining Resource Groups
z/OS (V1R3):	Chapter 7: Defining Resource Groups
z/OS (V1R4):	Chapter 7: Defining Resource Groups

Rule WLM010: Velocity goals have values which are too similar

Finding: CPExpert noticed that two or more service classes have been defined with an execution velocity goal and the Goal Importance was the same for both service classes, but that the actual values for the execution velocity goals were numerically similar. The Workload Manager cannot usually differentiate among execution velocities which are numerically similar.

Impact: This finding should be viewed as generally having a MEDIUM IMPACT on the performance of the service classes involved.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Installations may specify an *execution velocity goal* for a service class period. An execution velocity is a measure of how fast work should run when the work is ready to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage[1] The purpose of specifying an execution velocity goal is to allow installations to specify how important it is to have work processed, when the work has no time-related measure (that is, a response requirement is not associated with the work).

The execution velocity is computed based on samples collected at periodic sampling intervals[2] by the System Resources Manager (SRM). The SRM sampling code interrogates address space control blocks (TCBs, SRBs, OUCBs, and OUXBs) to determine the state of each address space assigned to a service class. Sampling counts associated with the service class are updated based upon the state[3] of the address spaces.

Please refer to Rule WLM104 for additional discussion of an execution velocity goal.

The sampling process takes samples every 250 milliseconds, resulting in 4 samples per second. The Workload Manager analyzes the results of the samples during the policy adjustment interval, which occurs every 10

---

[1]Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

[2]With MVS/ESA SP5.1, the sampling interval is 250 milliseconds. The state of each TCB or SRB associated with an address space is sampled every 250 milliseconds, beginning from address space initiation.

[3]Note that an address space can be in multiple states (for example, a CICS region might be using multiple processors concurrently, while some CICS tasks were also waiting on some function). Thus, the sample counts can total more than 100% of the sample intervals for any address space.

©Copyright 1995, Computer Management Sciences, Inc.

seconds. Consequently, the Workload Manager makes its policy adjustment decisions related to execution velocity, based on only 40 samples (4 samples per second * 10 seconds = 40 samples).

The margin of error based on 40 samples is sufficiently large that the process cannot differentiate between execution velocities which are numerically similar. If the execution velocity goals are numerically similar, the sampling error may be greater than the difference among the execution velocity goals.

IBM suggests that at least 10 should separate the goals (e.g., 10, 20, 30, etc.) for the Workload Manager to manage the system resources to meet the different execution velocity goals[4].

It is unfortunate that the *RMF Workload Activity Report* produced in Compatibility Mode lists execution velocity achieved as "xx.x" (for example, 61.3 execution velocity). Such a "precise" listing of the computed execution velocity misleads users into believing that the Workload Manager can manage system resources to achieve such detailed goals. Consequently, many early users of MVS/ESA SP5 (Goal Mode) have defined execution velocity goals with numerically similar values.

CPExpert produces Rule WLM010 when execution velocity goals are specified with less than 10 separating the goals, and the goals have the same Goal Importance.

The following example illustrates the output from Rule WLM010:

Suggestion: CPExpert suggests that you review the execution velocity goals specified for the service classes listed with Rule WLM010. You should revise the goals so that at least 10 separates each goal, or you should recognize that the Workload Manager cannot manage system resources to differentiate among the specified goals.

[4]This advice applies only to velocity goals with the same Goal Importance. During Policy Adjustment analysis, the Workload Manager examines service class periods in order of the Goal Importance. The service class with the highest Goal Importance is examined first. If the service classes have a different Goal Importance, the Goal Importance will provide the discrimination between the service classes and the execution velocity goals can be numerically similar without worry. Please see Section 4 (Chapter 3.5) for a discussion of the Policy Adjustment process.

```
RULE WLM010:  VELOCITY GOALS HAVE VALUES WHICH ARE TOO SIMILAR.

  CPExpert noticed that two or more service classes at the same Goal
  Importance have been defined with an execution velocity goal, but that
  the actual values for the execution velocity goals were numerically
  similar.  The Workload Manager cannot usually differentiate among
  execution velocities which are numerically similar.  At least 10 should
  separate the goals (e.g., 10, 20, 30, etc.) for the Workload Manager to
  manage the system resources to meet the different goals.  This finding
  applies to the following service classes:

           SERVICE   EXECUTION           SERVICE   EXECUTION
           CLASS     VELOCITY            CLASS     VELOCITY
           APPCFEED     50               ASCH         55
           APPCFEED     50               ST_TOOLS     57
           ASCH         55               MONITORS     50
           ASCH         55               ST_TOOLS     57
           ASCH         55               TPNSFEED     50
           MONITORS     50               ST_TOOLS     57
           ST_TOOLS     57               TPNSFEED     50
```

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

"MVS Workload Manager Velocity Goals: What you don't know can hurt you", John Arwe, IBM Corporation, CMG'96 Proceedings.

Rule WLM011: The service definition does not describe all workload

Finding: CPExpert noticed that not all work was classified using the workload classification scheme.

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the workload involved.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Installations establish workload classification rules to assign work to workload groups. A workload group is simply a "named" collection of similar work which has been categorized based upon classification rules[1].

In earlier versions of MVS, installations classified work using the IEAICSxx member of SYS1.PARMLIB. After applying classification rules in IEAICSxx, the resulting work would be assigned to performance groups.

With the Workload Manager, the same process conceptually applies. However, instead of assigning work to performance groups, the work is simply assigned to a "workload name" after applying classification rules and the workload name (or workload group name) is assigned to a service class.

A service class can represent any collection of workload that can be classified using the workload classification schemes available with the Workload Manager.

The Workload Manager provides default service classes for certain types of work:

- All system address spaces designated "high dispatching priority" address spaces are assigned to the **SYSTEM** Service Class. The high priority address spaces include MASTER, CRS, DUMPSRV, SMF, CATALOG, RASP, XCFAS, SMXC, CONSOLE, IOSAS, and others.

  Users **can** assign most[2] of these "high dispatching priority" address spaces to a different service class and provide the Workload Manager

[1]See IBM's *Planning: Workload Management* document cited in the Reference Section of this Rule Description for a complete discussion of the workload classification rules.

[2]The MASTER and INIT address spaces cannot be assigned to a service class other than the SYSTEM Service Class.

©Copyright 1995, Computer Management Sciences, Inc.

with performance goals for the address spaces. Assignment of these address spaces to a different service class would apply only in very unique situations.

- All started tasks not explicitly assigned to a service class by the workload classification scheme will be assigned to the **SYSSTC** Service Class.

- All other work not explicitly assigned to a service class by the workload classification scheme will be assigned to the **SYSOTHER** Service Class.

The SYSOTHER Service Class is assigned a discretionary performance goal by the Workload Manager. Consequently, any work that is assigned to the default SYSOTHER Service Class is processed on a discretionary basis by the Workload Manager.

CPExpert produces Rule WLM011 when the SYSOTHER Service Class appears in the SMF Type 72 (Subtype 3) records. Records for the SYSOTHER Service Class are produced only when workload executes in the SYSOTHER Service Class, indicating that some work was not classified by the workload classification scheme. CPExpert identifies the basic resources used by the SYSOTHER Service Class for each measurement interval in which the SYSOTHER records were available.

The following example illustrates the output from Rule WLM011:

```
RULE WLM011:  THE SERVICE DEFINITION DOES NOT DESCRIBE ALL WORKLOAD.

   CPExpert detected that some workload executed in the SYSOTHER service
   class.  The SYSOTHER service class is the default service class.  It is
   usually a good policy to explicitly define all workload, so you can be
   sure that you have properly assigned both performance goals and goal
   importance to the workload.  CPExpert will produce a report at the end
   of this run to identify the work units assigned to the SYSOTHER service
   class.
   WARNING: If the work defaulting to SYSOTHER is DDF enclaves, there will
            be no entries in the Type 30 records.  If CPExpert does not
            produce a report showing the work units assigned to SYSOTHER,
            you should check for DDF work that was not classified.
```

Suggestion: It is a good management practice to explicitly assign all work to a service class, thereby ensuring positive management control over all work. Even if the work would naturally fall into a service class with a discretionary performance goal, the workload classification scheme should be designed such that all work is explicitly identified. Only by explicitly identifying all work can management be sure that some important work is not "falling through the cracks" and being assigned a discretionary performance goal.

CPExpert will produce a report from the SMF Type 30 (Subtype 4) records to identify salient characteristics of the work units defaulting into the SYSOTHER Service Class. The report identifies the Subsystem Instance, the Job Name, Job Class, Program Name, Reader Start Time, and User Identification. This information should be sufficient to allow you to identify the work which is assigned by default to the SYSOTHER Service Class. You should modify your workload classification scheme to assign the work to an appropriate service class.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 10: Defining Classification Rules
OS/390 (V1R1):	Chapter 10: Defining Classification Rules
OS/390 (V1R2):	Chapter 10: Defining Classification Rules
OS/390 (V1R3):	Chapter 9: Defining Classification Rules
OS/390 (V2R4):	Chapter 9: Defining Classification Rules
OS/390 (V2R5):	Chapter 9: Defining Classification Rules
OS/390 (V2R6):	Chapter 9: Defining Classification Rules
OS/390 (V2R7):	Chapter 9: Defining Classification Rules
OS/390 (V2R8):	Chapter 9: Defining Classification Rules
OS/390 (V2R9):	Chapter 9: Defining Classification Rules
OS/390 (V2R10):	Chapter 9: Defining Classification Rules
z/OS (V1R1):	Chapter 9: Defining Classification Rules
z/OS (V1R2):	Chapter 9: Defining Classification Rules
z/OS (V1R3):	Chapter 9: Defining Classification Rules
z/OS (V1R4):	Chapter 9: Defining Classification Rules

Rule WLM012: A server defaulted to the SYSSTC Service Class

Finding: CPExpert noticed that a "server" address space defaulted to the SYSSTC Service Class.

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of the workload involved. However, the finding could have a HIGH IMPACT on performance during start-up of the server(s). Additionally, the finding could have a HIGH IMPACT on accounting for the use of system resources, on billing for use of system resources, or on capacity planning efforts.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class might be one of several "served" service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" might be related procurement, administration, miscellaneous, etc.).

©Copyright 1995, Computer Management Sciences, Inc.

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance normally are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager[1]. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class).

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICSPERS Service Class, as CICSPERS is simply a logical entity which describes transactions and CICSPERS is not an address space. Rather, the Workload Manager allocates resources to the "server" address space (the CICSRGN Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICSPERS Service Class, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" service classes which are served by the "server" service class. It does this by allocating resources to the "server" service class.

From an internal view, the Workload Manager establishes *internal service classes* for server service classes. These internal service classes are named **$SRMSnnn**, where "nnn" is simply an internal counter incremented based on the number of unique internal server service classes. The Workload Manager creates an internal $SRMSnnn service class for each unique set of address spaces that serve the same set of served service classes. It is to these internal $SRMSnnn service classes that the Workload Manager assigns resources, based on the performance of the served service classes.

For example, one or more CICS regions would be described by a single internal $SRMSnnn service class, if the CICS regions provide service to the same set of served service classes. To illustrate, a CICS Terminal Owning Region (TOR) and a CICS Application Owning Region (AOR) may provide service to one or more service classes which describe CICS transactions.

The Workload Manager will make resource allocation decisions for the TOR and AOR based on the performance of the CICS transactions. The

[1]This statement is not **strictly** true. If the CICS region should become idle for an extended period (no transactions executed in the "served" service classes), the Workload Manager would use the service goal and importance specified for the CICS region service class to manage the region. Practically, of course, there would be little to manage with an idle region.

TOR and AOR address spaces will be assigned to an internal $SRMSnnn service class, and the resource allocation decisions would be made relative to the $SRMSnnn service class.

In our example, the TOR and AOR regions may be not be explicitly described by the workload classification scheme. In this case (assuming that the regions are started as started tasks), the TOR and AOR would be assigned by default to the SYSSTC Service Class, from an external view. The TOR and AOR would be managed internally by the $SRMSnnn service class as described above.

The resources used by the TOR and AOR will be combined with other started tasks executing in the SYSSTC Service Class, and reported in SMF Type 72 (Subtype 3) records. Once the resources used by the TOR and AOR are combined with other started tasks into the SYSSTC Service Class, it is impossible to account for the resources used by the TOR and AOR. Consequently, accounting and capacity planning efforts may be unsuccessful.

During the start-up of the server address spaces, the servers would be assigned the dispatching priority of the SYSSTC service class. The dispatching priority of the SYSSTC service class is very high. Start-up of the server address spaces could use a relatively significant amount of CPU time, executing at the high SYSSTC dispatching priority[2]. Consequently, performance of other service classes could be seriously degraded during start-up of the server address spaces.

CPExpert detects whether the SYSSTC Service Class provides service to any service classes. This detection is done by determining whether any elements appear in the "Service Class Served Data Section" for the SYSSTC Service Class.

CPExpert produces Rule WLM012 if any elements appear in the "Service Class Served Data Section" for the SYSSTC Service Class.

It is possible, of course, to assign the TOR and AOR to *report classes* in the workload classification scheme. In this case, the resources used by the TOR and AOR are identified and reported by the report class to which they are assigned. Unfortunately, CPExpert cannot determine whether server service classes have been assigned to report classes.

The following example illustrates the output from Rule WLM012:

[2]Address spaces in SYSSTC service class execute at dispatching priority FD (253) if APAR OW19265 is **not** applied, and execute at dispatching priority of FE (254) if OW19265 **is** applied.

```
RULE WLM012:  A SERVER DEFAULTED TO THE SYSSTC SERVICE CLASS

  CPExpert noticed that a "server" address space defaulted to the SYSSTC
  service class.  Address spaces in the SYSSTC service execute at a very
  high dispatching priority (e.g., 253).  Performance of other service
  classes could be significantly degraded during start-up of the server
  service class, as a relatively large amount of CPU time could be  used
  at a high dispatching priority.  A server address space (a CICS region,
  IMS region, etc.) normally should be explicitly assigned to a service
  class.
```

Suggestion: CPExpert suggests that you review the workload classification scheme to determine which "server" address spaces have been allowed to be assigned by default to the SYSSTC Service Class. CPExpert suggests that the workload classification scheme be adjusted to assign the "server" address spaces (1) to their own service class or (2) to a report class. There are several significant reasons for this suggestion:

- Performance of other service classes could be seriously degraded during start-up of the server address spaces.

- Storage isolation is not invoked for SYSSTC started tasks unless cross memory page faults are affecting other work with goals. Storage isolation can improve the performance of servers (e.g., CICS or IMS) at many sites.

- Management reporting will be misleading if the resources used by the servers are grouped into the SYSSTC Service Class.

- Resource accounting and billing will be unable to determine the resources used by the servers and thus will be unable to allocate resource use to their corresponding transactions. Consequently, resource accounting and billing will be unable to apportion the resource use to organization entities or applications using the resources.

- Capacity planning efforts will not be able to identify the resources used by the type of transactions, and consequently capacity planners will have insufficient information.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

MVS Programming: Workload Management Services

MVS/ESA(SP 5):	Chapter 4: Using SMF Record Type 99
OS/390 (V1R1):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R2):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R3):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R4):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R5):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R6):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R7):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R8):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R9):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R10):	Chapter 10: Using SMF Record Type 99
z/OS (V1R1):	Chapter 10: Using SMF Record Type 99
z/OS (V1R2):	Chapter 10: Using SMF Record Type 99
z/OS (V1R3):	Chapter 10: Using SMF Record Type 99
z/OS (V1R4):	Chapter 10: Using SMF Record Type 99

"WLM Server Management", Peter Enrico, SHARE 1999 Summer Conference.

Rule WLM013: Response goal was specified for server service class

Finding: CPExpert noticed that a response goal was specified for a service class that was a "server" service class.

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of the workload involved.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class may be one of several "served" service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" may be related procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by

the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class)[1].

Of particular importance to this finding is the fact that the Workload Manager detects that a service class is not meeting its response goal based on the response experienced by a work element (either a job or a transaction) executing in the service class. The Workload Manger can measure the response of a job or transaction only when the job or transaction ends. Thus, the Workload Manger detects that the service class is not meeting its response performance goal based on the response experienced by **previously-ended** jobs or transactions.

Address spaces in a "server" service class do not report an "ended transaction" until the address space ends[2]. Once the address space ends, there is no further policy adjustment necessary by the Workload Manager. Consequently, a response goal will be completely ineffective for a server service class.

CPExpert produces Rule WLM013 when a response goal is specified for a server service class.

The following example illustrates the output from Rule WLM013:

```
RULE WLM013:  RESPONSE GOAL WAS SPECIFIED FOR SERVER SERVICE CLASS.

   CPExpert noticed that a response performance goal was specified for the
   CICSTEST Service Class.  This service class is a "server" service class.
   The performance goal of the server is used only at address space start-up
   and termination.  The performance goal is not used by the Workload Manger
   to evaluate the performance of the transactions served by the server.
   You should specify an execution velocity goal for the server.
```

Suggestion: CPExpert suggests that you specify an execution velocity performance goal for the server service class identified by this rule.

---

[1]This statement is not **strictly** true. If the CICS region should become idle for an extended period (no transactions executed in the "served" service classes), the Workload Manager would use the service goal and importance specified for the CICS region service class to manage the region. Practically, of course, there would be little to manage with an idle region.

[2]There is an exception to this, in case the System Resources Manager (SRM) detects that internal variables are in danger of overflowing. If the SRM detects a danger of overflow for an address space, the SRM will "end" the address space internally and "restart" the address space. Consequently, there may be "ended transactions" for a server address space even though the address space has not ended from an external view. These situations should be extremely rare and normally occur only if the MSO service coefficient is set to a value greater than one or in cases of "never-ending" address spaces.

Practically, the performance goal has little effect since it applies only at address space start-up. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes.

However, you may wish to specify a relatively high execution velocity goal to ensure rapid start-up of the CICS region.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 12: Defining Service Units and Coefficients
OS/390 (V1R1):	Chapter 12: Defining Service Units and Coefficients
OS/390 (V1R2):	Chapter 12: Defining Service Units and Coefficients
OS/390 (V1R3):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R4):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R5):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R6):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R7):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R8):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R9):	Chapter 11: Defining Service Units and Coefficients
OS/390 (V2R10):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R1):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R2):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R3):	Chapter 11: Defining Service Units and Coefficients
z/OS (V1R4):	Chapter 11: Defining Service Units and Coefficients

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

©Copyright 1995, Computer Management Sciences, Inc.

Rule WLM014: A response goal was specified for "hot batch" workload

Finding: CPExpert noticed that a response goal was specified for a service class which might describe "hot batch" workload.

Impact: This finding should be viewed as generally having a MEDIUM IMPACT or HIGH IMPACT on the performance of the workload involved.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Many organizations have a "hot batch" category of work, which describes work that is suddenly submitted to Operations with an immediate priority for execution and successful completion. Such a category might apply to critical work submitted by executive level management, work which has critical deadlines but which earlier failed to complete, etc. Whatever the reason, a "hot batch" category of work typically exists in most organizations.

By its normal definition, jobs do not frequently nor regularly appear in the "hot batch" category. That is, normally there is not a large number of "hot batch" jobs executing on the system, nor is there a continual supply of "hot batch" jobs submitted for execution. The "hot batch" category usually is reserved only for those infrequent batch jobs which **must** be completed as soon as possible. Consequently, the "hot batch" service class would have a very high priority relative to service classes describing other batch work.

Some organizations assign a response performance goal to the "hot batch" jobs, with a small value for the goal. The organizations believe that the small response goal will ensure that the "hot batch" job will be given a high priority by the Workload Manager and that the Workload Manager will manage system resources in an attempt to meet the goal.

Unfortunately, a response goal often will not achieve the desired result.

The Workload Manger detects that a service class is not meeting its response goal based on the response experienced by work elements (either jobs or transactions) executing in the service class. The Workload Manger can measure the response of a job or transaction only when the job or transaction ends. Thus, the Workload Manger detects that the service class is not meeting its response performance goal based on the response experienced by **previously-ended** jobs or transactions.

©Copyright 1995, Computer Management Sciences, Inc.

Additionally the Workload Manager retains "historical" information for only 20 minutes. The 20-minute limit was implemented because the Workload Manager developers felt that data over 20 minutes old would represent "old" information and should not guide "current" Workload Manager decisions[1]

If "hot batch" jobs occur relatively infrequently, there would no immediate history of response time, and there may be no history at all.
The Workload Manager could not make any policy adjustment decisions relative to the response performance goal for the "hot batch" service class, since no recent response data would be available. Consequently, a response performance goal is inappropriate for a "hot batch" service class.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the phrase "hot batch" and assumes that the service class describes a "hot batch" workload if "hot batch" is encountered. Additionally, CPExpert examines the service class name (SMF Type 72 field R723MCNM) to see whether "HOTBATCH" is specified as the name of the service class. There are two considerations with this approach:

- It is, of course, possible that the service class does not describe a "hot batch" workload even though "hot batch" is in the description or in its name. This instance is unlikely, as most installations will use the phrase "hot batch" in a description of only "hot batch" work.

- It also is possible that "hot batch" will not be in the description of a service class describing "hot batch" workload. CPExpert will be unable to detect these situations.

If CPExpert detects "hot batch" in the Service Class Description, CPExpert examines the performance goal to see whether the goal type is a response goal (either an average response or a percentile response).

CPExpert produces Rule WLM014 when a response goal is specified for a service class that potentially describes "hot batch" workload.

The following example illustrates the output from Rule WLM014:

[1]As described in Section 4, the Workload Manager makes decisions every 10 seconds about whether to adjust the resource policies. In 20 minutes, the Workload Manager would have made 120 such policy adjustment decisions. It is unlikely that "old" information after 120 policy adjustment decisions would still be relevant.

```
RULE WLM014:  RESPONSE GOAL WAS SPECIFIED FOR "HOT BATCH" WORKLOAD.

   CPExpert noticed that a response goal was specified for a service class
   that had "hot batch" as a part of its description.  Many installations
   use "hot batch" to describe batch jobs which infrequently are submitted,
   but which require rapid turnaround.  Unfortunately, a response goal may
   not achieve the desired effect.  Please refer to the WLM Component User
   Manual for a discussion of this issue.  If the considerations described
   in the User Manual do not apply, you should ignore this finding. This
   finding applies to the following service class:

            SERVICE CLASS    PERIOD    DESCRIPTION
            HOTBATCH            1      Special hot batch applications
```

Suggestion: CPExpert suggests that you review the service class identified by Rule WLM014. If the service class does describe "hot batch" workload, CPExpert suggests that you change the response performance goal to an execution velocity goal.

As described above, it is unlikely that the Workload Manager will be able to make policy adjustment decisions to achieve the response goal for the "hot batch" service class. This is because the Workload Manager does not have response information until the job completes; too late to help performance of the "hot batch" job being processed!

However, the Workload Manager collects CPU using and delay information on a continual basis[2]. You can assign a high execution velocity performance goal to the service class describing "hot batch" workload, and the Workload Manager will have immediate information upon which to make policy adjustment decisions.

You should, of course, make sure that the execution velocity performance goal you establish for the "hot batch" service class is consistent with the importance of work executing in the service class.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals

[2]Please refer to Section 4 for a description of the sampling process used to acquire information necessary to compute execution velocity.

OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

Rule WLM015: Execution velocity goal specified for TSO Period 1 or Period 2

Finding: An execution velocity goal was specified for TSO Period 1 or Period 2.

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the service class periods involved. The level of impact depends on the value specified for the execution velocity.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Installations may specify an *execution velocity goal* for a service class period. An execution velocity is a measure of how fast work should run when the work is ready to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage[1] The purpose of specifying an execution velocity goal is to allow installations to specify how important it is for work to have access to a CPU, when the work has no time-related measure (that is, a response requirement is not associated with the work).

The execution velocity is computed based on samples collected at periodic sampling intervals[2] by the Workload Manager (SRM). The SRM sampling code interrogates address space control blocks (TCBs, SRBs, OUCBs, and OUXBs) to determine the state of each address space assigned to a service class. Sampling counts associated with the service class are updated based upon the state[3] of the address spaces.

Please refer to Rule WLM104 for additional discussion of an execution velocity goal.

A service class representing TSO Period 1 or TSO Period 2 transactions normally has a *response goal* defined. This is because these TSO transactions normally are quite interactive, and response time is an appropriate measure of performance. These transactions may be contrasted with TSO Period 3, for example. TSO Period 3 typically contains non-interactive transactions, even though the transactions were

---

[1]Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

[2]With MVS/ESA SP5.1, the sampling interval is 250 milliseconds. The state of each TCB or SRB associated with an address space is sampled every 250 milliseconds, beginning from address space initiation.

[3]Note that an address space can be in multiple states (for example, a CICS region might be using multiple processors concurrently, while some CICS tasks were also waiting on some function). Thus, the sample counts can total more than 100% of the sample intervals for any address space.

©Copyright 1995, Computer Management Sciences, Inc.

submitted under TSO. Since TSO Period 3 normally represents non-interactive transactions, a response goal is inappropriate for TSO Period 3. Instead of a response goal, either an execution velocity goal or a discretionary goal is usually specified for TSO Period 3.

When an execution velocity goal is specified for TSO Period 1 or TSO Period 2, the Workload Manager determines whether the TSO transactions *receive* CPU service when they *require* CPU service. Thus, the goal against which performance is being evaluated is **resource oriented** rather than **response time oriented**.

Users, of course, are not as concerned with resource performance as they are concerned with response time. Consequently, service agreements normally are written in terms of response time. Response time normally should be the goal against which the Workload Manager is measuring performance for TSO Period 1 or Period 2.

There are several system-related disadvantages to specifying a velocity goal for TSO Period 1 or Period 2:

- The SRM controls swap protect time on an **individual address space** basis for service classes with an execution velocity goal, rather than controlling swap protect time on a **service class** basis as is done for response goals. Controlling swap protect time on an individual address space basis adds unnecessary overhead.

- If the TSO transactions take awhile to complete (and an execution velocity goal has been specified), the SRM will examine TSO address spaces to decide what expanded storage access to give to their demand paging, VIO paging, and hiperspace paging.

- The SRM might implement working set management algorithms to monitor the TSO address space if an execution velocity goal is specified for the TSO service class. This overhead is not required if a response goal is assigned to the TSO service class.

CPExpert produces Rule WLM015 when an execution velocity goal was specified for TSO Period 1 or Period 2.

The following example illustrates the output from Rule WLM015:

```
RULE WLM015:  EXECUTION VELOCITY GOAL SPECIFIED FOR TSO PERIOD 1 OR 2.

 CPExpert noticed that an execution velocity performance goal was specified
 for a service class (Period 1 or Period 2) that had "TSO" as a part of
 its description.  Under most circumstances, you should specify a response
 goal for TSO Period 1 or Period 2.  Not only is it more efficient from a
 Workload Manager view to specify a response goal for TSO, but users tend
 to relate well to response time (rather than execution velocity).  There
 are a few situations in which you may wish to specify an execution
 velocity goal for TSO Period 1 or Period 2.  These are described in the
 WLM Component User Manual.  Unless these unusual situations apply in
 your environment, CPExpert suggests that you change the performance goal
 from an execution velocity goal to a response goal for the TSO Service
 Class (Period 1).
```

Suggestion: CPExpert suggests that you review the execution velocity goal specified for the service class periods listed with Rule WLM015. CPExpert suggests that you consider changing the goals to specify a response goal for the TSO service class periods identified by this rule.

There are some situations in which you may wish to specify an execution velocity goal for a TSO Period 1 or Period 2 service class. **If these situations apply, please ignore the above suggestion**.

- You may have established a service class for TSO transactions, but expect to have very few transactions execute in the service class. If few transactions execute in a service class, the Workload Manager would not have any "history" of response time from which to compute average response time or meaningful percentile response time. If the Workload Manager cannot compute the average or percentile response time, insufficient information would be available for the Workload Manager to make its decisions. In such a situation, you may have specified an execution velocity simply to ensure that the TSO transactions received some level of service.

- You may have an unusual situation in which you have established a service class for TSO transactions which typically take a long time to run. Some organizations use TSO to perform functions that normally might be considered batch jobs. If you have such a situation, you may have specified a TSO service class with an execution velocity for either Period 1 or Period 2. Since a response goal would be inappropriate, the execution velocity goal may be an appropriate specification.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Effective Use of MVS Workload Manager Controls", Ed Berkel and Peter Enrico, IBM Corporation, CMG Transactions, Winter 1995, the Computer Measurement Group.

Rule WLM016: Low velocity goal was specified for server service class

Finding: CPExpert noticed that a low velocity goal was specified for a service class that was a "server" service class.

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of the workload involved. This finding may have a HIGH IMPACT during address space start-up or shutdown.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class may be one of several "served" service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" may be related procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance normally are used by the Workload Manager **only at address space start-up** time. After the

©Copyright 1995, Computer Management Sciences, Inc.

CICS region has started, its performance goals and importance **normally** are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class)[1].

Of particular importance to this finding is the fact that the Workload Manager **does** use the performance goal for the server service class during address space start-up (and may use it during address space shutdown), and may use the performance goal if the server should become idle for an extended period of time. The actual value specified for the goal **is important** during these intervals. You may experience serious performance problems during start-up of a CICS region (and potentially during region shutdown) if an inappropriate goal is specified for the server service class. Consequently, you should not specify a low execution velocity for the service class describing a server.

CPExpert produces Rule WLM016 when less than 30% was specified as an execution velocity goal for a **server** service class.

The following example illustrates the output from Rule WLM016:

```
RULE WLM016:  LOW EXECUTION VELOCITY GOAL SPECIFIED FOR SERVER.

  CPExpert noticed that a relatively low execution velocity was specified
  for a service class that was a "server" for subsystem transactions (that
  is, the service class is a CICS region or IMS region).  A low execution
  velocity for a server may delay the start-up or shutdown of the address
  space.  You may wish to specify a higher execution velocity for the
  CICSTEST Service Class.
```

Suggestion: CPExpert suggests that you specify a larger execution velocity performance goal for the server service class identified by this rule.

As mentioned above, the performance goal has little effect since it applies only at address space start-up. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes.

[1]If the CICS region should become idle for an extended period (no transactions executed in the "served" service classes), the Workload Manager would use the service goal and importance specified for the CICS region service class to manage the region. Practically, of course, there would be little to manage with an idle region.

You may wish to specify a relatively high execution velocity goal to ensure rapid start-up of the CICS region. This relatively high execution velocity goal should be specified even if the transactions served by the server might be assigned a relatively low response goal.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

©Copyright 1995, Computer Management Sciences, Inc.

Rule WLM017: Server and subsystem transactions in same service class

Finding: The policy being evaluated contained a service class into which the workload classification scheme had placed both a server (e.g., a CICS region) and the transactions being served by the server (e.g., the CICS transactions).

Impact: This finding should be viewed as generally having a LOW IMPACT on the performance of the workload involved. This finding may have a HIGH IMPACT during address space start-up or shutdown.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

Please refer to Rule WLM016 for discussion of the relationship between server service classes and transaction service classes.

Of particular importance to this finding is the fact that the Workload Manager **uses the performance goal for the server service class** during address space start-up (and may use it during address space shutdown). The actual value specified for the goal **is important** during these intervals. You may experience serious performance problems during start-up of a CICS region (and potentially during region shutdown) if an inappropriate goal is specified for the server service class.

The server service class consists of address spaces, while the subsystem transaction service classes consists of transactions. The address spaces do not have response goals, since an address space "transaction" ends (from the view of the SRM) only when the address space ends. On the other hand, the subsystem transactions do have response goals. Further, since the transaction service classes are simply logical groupings of transactions and not address spaces, the transaction service classes do not have resource use associated with them (the transactions do not use resources; the address spaces use resources).

©Copyright 1995, Computer Management Sciences, Inc.

It is illogical to specify the same performance goal for a server and the transactions it serves; the server goal should relate to resources and the transaction goal should relate to response time. Consequently, you should not group transactions into the same service class as the address spaces serving the transactions.

With OS/390 V2R10, IBM introduced an “exemption from transaction response time management” option. This option is available with APAR OW43812 installed. With the APAR applied, organizations can specify whether an address space (CICS region or IMS region) will be managed based on the goals of the transactions that the region is serving, or managed based on the goals specified for the region itself. This option is exercised by using the new “Manage Region Using Goals Of:” field on the WLM ISPF “Modify Rules for the Subsystem Type” panel.

When “TRANSACTION” is entered in the “Manage Region Using Goals OF:” field, the region will be managed as a CICS/IMS transaction server by the WLM. “TRANSACTION” is the default specification. If “REGION” is entered in this field, the region will be managed based on the performance goal specified for the service class to which the region is assigned. This performance goal normally would be an execution velocity goal.

When “REGION” is specified, the WLM does not consider the region to be a “server” of transactions. Rather, the WLM server topology algorithms ignore the region when establishing server topology. Consequently, the goals for any transaction processed by the region will not be considered by the WLM when it determines whether service class periods meet goals and whether policy adjustment is necessary.

This consequence might have undesired implications if you specify goals for CICS or IMS transactions and some or all of those transactions are processed by a CICS or IMS region that has “REGION” specified in the “Manage Region Using Goals Of:” field. In this case, **performance of the transaction service class will not be considered when adjusting resource policy for the region**. This could have the undesired result of transactions not achieving the performance that you desire, simply because the transactions were processed by a CICS or IMS region that was managed based on the goals specified for the region. Alternatively, some transactions might receive better performance than desired because of the same “region-oriented” management by the WLM.

CPExpert produces Rule WLM017 when a service class has been identified as a server, and the SMF record contains Work Manager/Resource Manager State data.

The following example illustrates the output from Rule WLM017:

```
RULE WLM017:  SERVER AND TRANSACTIONS ARE IN SAME SERVICE CLASS

   CPExpert noticed that the CICSDEFA Service Class was defined to include
   both an address space providing service (e.g., a CICS region) and the
   subsystem transactions being served by the address space.  CPExpert
   suggests that you place the server address space(s) and the subsystem
   transactions being served into different service classes.
```

Suggestion: CPExpert suggests that you change your workload classification scheme to place server address spaces into a different service class than the subsystem transactions being served.

- An appropriate execution velocity goal (greater than 30) should be defined for the **server** service class.

- An appropriate response goal should be defined for the **transaction** service class(es).

Alternatively, if you are running OS/390 Version 2 Release 10 (with APAR OW43812 installed) you might have deliberately assigned the “served” transaction service class to the region, and specified that the region be managed based on the goals specified for the region (normally, this would be an execution velocity goal). If this is the situation, you should consider the following alternatives:

- You can simply ignore this finding, but allow CPExpert to continue to check for such situations. The reason that you might wish to allow CPExpert continue to invoke Rule WLM017 is that your workload classification might change, new transactions might be added, CICS or IMS might route transactions to the CICS region or message processing region, etc. You might not be aware of the implications of the WLM assigning a specific transaction to a region managed by the region’s goal.

- You can “turn off” Rule WLM017 as described in Section 2 of this WLM Component User Manual. This action should be taken if you become annoyed by Rule WLM017 being produced when you do not plan to take action.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

Rule WLM018: Multiple periods specified for server service class

Finding: CPExpert noticed that more than one period had been specified for a server service class.

Impact: This finding has a LOW impact on the performance of your system. The impact mostly results from increased (and unnecessary) overhead.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: A service class may be broken into multiple service class **periods**. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional service class periods (although typically no more than two or three additional service class periods are defined). Each service class period can have its own performance goals, defined to the Workload Manager via the Workload Manager ISPF panels.

An address space (TSO transaction, batch job step, etc.) begins in Period 1 of the service class to which it is assigned. The address space transitions from Period 1 to Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction. The "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources.

The normal purpose of defining multiple service class periods is to give higher importance to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources do not compete for system resources with those address spaces requiring substantial resources[1].

If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

[1]Note that with MVS/ESA SP5.1, the redesign of the MVS Dispatcher algorithms eliminated much of the potential conflict for CPU cycles (although other considerations apply). Please refer to Rule WLM025 for additional discussion in this area.

©Copyright 1995, Computer Management Sciences, Inc.

Please refer to Section 4 for discussion of the relationship between server service classes and transaction service classes.

Of particular importance to this finding is the fact that the Workload Manager normally uses the performance goal for the server service class only during address space start-up (and may use it during address space shutdown). During other times, the Workload Manager **uses the performance goals (and goal importance) of the transaction service classes being served**, and ignores the performance goal specified for the server.

Since the only purpose of having multiple periods for a service class is to specify different performance goals and/or goal importance for different categories of work, and since the Workload Manager ignores the performance goal and goal importance of the server, multiple periods should not be specified for server service classes. The changed performance goal or goal importance will be ignored by the Workload Manager, and the server will not "migrate" to lower service class periods. However, the system will incur additional overhead caused by the unused service class period(s). As examples of the overhead:

- Additional SRM control blocks are created and processed.
- Additional Workload Manager control blocks are created and processed.
- RMF requires additional processing of the control blocks.
- SMF Type 72(Subtype 3) records are written for each service class period defined (regardless of whether the service class periods are used by the Workload Manager).

CPExpert produces Rule WLM018 when it detects that multiple periods are defined for a server service class.

The following example illustrates the output from Rule WLM018:

```
RULE WLM018:  MULTIPLE PERIODS WERE SPECIFIED FOR SERVER

   CPExpert noticed that the CICSRGN Service Class was defined to include
   multiple periods.  The CICSRGN Service Class is a server (e.g., a CICS
   or IMS region), and multiple periods cannot be used for servers.
   CPExpert suggests that you eliminate the multiple-period definition
   for the CICSRGN Service Class, since the unused periods will generate
   unnecessary overhead.
```

Suggestion: CPExpert suggests that you eliminate the multiple periods for the server service class. These periods are not used, but they cause unnecessary overhead.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8.2: Using Performance Periods
OS/390 (V1R1):	Chapter 8.2: Using Performance Periods
OS/390 (V1R2):	Chapter 8.2: Using Performance Periods
OS/390 (V1R3):	Chapter 8.2: Using Performance Periods
OS/390 (V2R4):	Chapter 8.3: Using Performance Periods
OS/390 (V2R5):	Chapter 8.3: Using Performance Periods
OS/390 (V2R6):	Chapter 8.3: Using Performance Periods
OS/390 (V2R7):	Chapter 8.3: Using Performance Periods
OS/390 (V2R8):	Chapter 8.3: Using Performance Periods
OS/390 (V2R9):	Chapter 8.3: Using Performance Periods
OS/390 (V2R10):	Chapter 8.3: Using Performance Periods
z/OS (V1R1):	Chapter 8.3: Using Performance Periods
z/OS (V1R2):	Chapter 8.3: Using Performance Periods
z/OS (V1R3):	Chapter 8.3: Using Performance Periods
z/OS (V1R4):	Chapter 8.3: Using Performance Periods

©Copyright 1995, Computer Management Sciences, Inc.

Rule WLM019: Multiple periods specified for subsystem transaction service class

Finding: Multiple periods have been specified for a subsystem transaction service class.

Impact: This finding has a LOW impact on the performance of your system. The impact mostly is a result of increased (and unnecessary) overhead.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: A service class may be broken into multiple service class **periods**. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional service class periods (although typically no more than two or three additional service class periods are defined). Each service class period can have its own performance goals, defined to the Workload Manager via the Workload Manager ISPF panels.

An address space (TSO transaction, batch job step, etc.) begins in Period 1 of the service class to which it is assigned. The address space transitions from Period 1 to Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction. The "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources.

The normal purpose of defining multiple service class periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources[1].

If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

[1]Note that with MVS/ESA SP5.1, the redesign of the MVS Dispatcher algorithms eliminated much of the potential conflict for CPU cycles (although other considerations apply). Please refer to Rule WLM025 for additional discussion in this area.

©Copyright 1995, Computer Management Sciences, Inc.

Please refer to Section 4 for discussion of the relationship between server service classes and transaction service classes.

Of particular importance to this finding is the fact that the System Resources Manager (SRM) **does not accumulate resources** for subsystem transaction service classes. These transaction service classes are simply logical groupings of transactions based on the workload classification scheme, and the actual resources are used by (and attributed to) the address spaces processing the transactions.

For example, CICS transactions may be grouped into a transaction service class. CICS/ESA Version 4 reports information about the transactions to the Workload Manager (transaction response time and transaction delays). The CICS region is the address space which executes CICS tasks to process the CICS transactions, and the region address space uses resources.

Since an address space transitions from Period 1 to Period 2 (and to subsequent periods) based upon the accumulation of "service" by the transaction, and since the transactions are not address spaces, the transactions cannot transition from Period 1 to Period 2. Consequently, any period other than Period 1 is simply unused. However, the system will incur additional overhead caused by the unused service class period(s). As examples of the overhead:

- Additional SRM control blocks are created and processed.
- Additional Workload Manager control blocks are created and processed.
- RMF requires additional processing of the control blocks.
- SMF Type 72(Subtype 3) records are written for each service class period defined (regardless of whether the service class periods are used by the Workload Manager).

CPExpert produces Rule WLM019 when it detects that multiple periods are defined for a service class describing subsystem transactions (e.g., CICS or IMS transactions).

The following example illustrates the output from Rule WLM019:

```
RULE WLM019:  MULTIPLE PERIODS WERE SPECIFIED FOR SUBSYSTEM TRANSACTIONS

   CPExpert noticed that the CICSDEFA Service Class was defined to include
   multiple periods.  The CICSDEFA Service Class consists of subsystem
   transactions (e.g., CICS or IMS transactions), and multiple periods
   cannot be used for subsystem transactions.  CPExpert suggests that you
   eliminate the multiple-period definition, since the unused periods will
   generate unnecessary overhead.
```

Suggestion: CPExpert suggests that you eliminate the multiple periods for the transaction service class. These periods are not used, but they cause added overhead.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8.2: Using Performance Periods
OS/390 (V1R1):	Chapter 8.2: Using Performance Periods
OS/390 (V1R2):	Chapter 8.2: Using Performance Periods
OS/390 (V1R3):	Chapter 8.2: Using Performance Periods
OS/390 (V2R4):	Chapter 8.3: Using Performance Periods
OS/390 (V2R5):	Chapter 8.3: Using Performance Periods
OS/390 (V2R6):	Chapter 8.3: Using Performance Periods
OS/390 (V2R7):	Chapter 8.3: Using Performance Periods
OS/390 (V2R8):	Chapter 8.3: Using Performance Periods
OS/390 (V2R9):	Chapter 8.3: Using Performance Periods
OS/390 (V2R10):	Chapter 8.3: Using Performance Periods
z/OS (V1R1):	Chapter 8.3: Using Performance Periods
z/OS (V1R2):	Chapter 8.3: Using Performance Periods
z/OS (V1R3):	Chapter 8.3: Using Performance Periods
z/OS (V1R4):	Chapter 8.3: Using Performance Periods

Rule WLM020: Subsystem transactions in same service class as address space

Finding: The policy being evaluated contained a service class into which the workload classification scheme had placed subsystem transactions (e.g., CICS transactions) into the same service class as were placed address spaces (e.g., TSO transactions).

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the workload involved.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The Workload Manager allows installations to group similar work into service classes. For example, TSO transactions might be placed into one service class, batch jobs might be placed into another service class, etc. Users define service classes by specifying the name of the service class and specifying the performance goal, service class, and service class importance.

If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

Please refer to Section 4 for discussion of the relationship between server service classes and transaction service classes.

The Workload Manager ISPF application does not prevent an installation from grouping subsystem transactions into the same service class as address spaces. For example, an installation may define a workload classification scheme to place certain TSO transactions and certain CICS into an *INTERACT Service Class*. The workload classification algorithms would dutifully place the TSO transactions (really address spaces) and the CICS transactions (logical entities which would be processed by CICS tasks), into the INTERACT Service Class.

There are many disadvantages to such a workload classification scheme and service class structure. As examples:

- If a response goal were specified for the INTERACT Service Class, the Workload Manager would use the response time of both TSO transactions and CICS transactions as the basis for analyzing whether the service class met its performance goal. The TSO transactions and the CICS transactions typically would have vastly different response characteristics.

- If multiple periods were defined for the INTERACT Service Class, only TSO transactions could migrate to any period other than Period 1. Resources are not accumulated for subsystem transactions and they are not address spaces, so they would not migrate to Period 2 and beyond.

- If the example INTERACT Service Class did not meet its performance goal, the Workload Manager must decide whether to allocate resources to the INTERACT Service Class (for the TSO transactions) **or to the server service class** (the CICS region providing service to the CICS transactions).

The bottom line is that such a workload classification scheme is "mixing apples and oranges" and would not yield good results.

With OS/390 V2R10, IBM introduced an “exemption from transaction response time management” option. This option is available with APAR OW43812 installed. With the APAR applied, organizations can specify whether an address space (CICS region or IMS region) will be managed based on the goals of the transactions that the region is serving, or managed based on the goals specified for the region itself. This option is exercised by using the new “Manage Region Using Goals Of:” field on the WLM ISPF “Modify Rules for the Subsystem Type” panel.

When “TRANSACTION” is entered in the “Manage Region Using Goals OF:” field, the region will be managed as a CICS/IMS transaction server by the WLM. “TRANSACTION” is the default specification. If “REGION” is entered in this field, the region will be managed based on the performance goal specified for the service class to which the region is assigned. This performance goal normally would be an execution velocity goal.

When “REGION” is specified, the WLM does not consider the region to be a “server” of transactions. Rather, the WLM server topology algorithms ignore the region when establishing server topology. Consequently, the goals for any transaction processed by the region will not be considered by the WLM when it determines whether service class periods meet goals and whether policy adjustment is necessary.

This consequence might have undesired implications if you specify goals for CICS or IMS transactions and some or all of those transactions are

processed by a CICS or IMS region that has “REGION” specified in the “Manage Region Using Goals Of:” field. In this case, **performance of the transaction service class will not be considered when adjusting resource policy for the region**. This could have the undesired result of transactions not achieving the performance that you desire, simply because the transactions were processed by a CICS or IMS region that was managed based on the goals specified for the region. Alternatively, some transactions might receive better performance than desired because of the same “region-oriented” management by the WLM.

CPExpert produces Rule WLM020 when a service class has been identified as an address space (the service class has accumulated service), and the SMF record contains Work Manager/Resource Manager State data. Rule WLM020 is suppressed if the address space is a server, as Rule WLM017 addresses this issue.

The following example illustrates the output from Rule WLM020:

```
RULE WLM020:  SUBSYSTEM TRANSACTIONS IN SERVICE CLASS WITH ADDRESS SPACE

   CPExpert noticed that the INTERACT Service Class was defined to include
   both subsystem transactions (e.g., CICS transactions) and one or more
   address spaces accumulating service (e.g., TSO transactions).  CPExpert
   suggests that your workload classification scheme be revised to place
   the subsystem transactions into a different service class than the
   address spaces.
```

Suggestion: CPExpert suggests that you change your workload classification scheme to place the subsystem transactions into a different service class from the address spaces.

Alternatively, if you are running OS/390 Version 2 Release 10 (with APAR OW43812 installed) you might have deliberately assigned the “served” transaction service class to the region, and specified that the region be managed based on the goals specified for the region (normally, this would be an execution velocity goal). If this is the situation, you should consider the following alternatives:

- You can simply ignore this finding, but allow CPExpert to continue to check for such situations. The reason that you might wish to allow CPExpert continue to invoke Rule WLM020 is that your workload classification might change, new transactions might be added, CICS or IMS might route transactions to the CICS region or message processing region, etc. You might not be aware of the implications of the WLM assigning a specific transaction to a region managed by the region’s goal.

- You can “turn off” Rule WLM020 as described in Section 2 of this WLM Component User Manual. This action should be taken if you become annoyed by Rule WLM020 being produced when you do not plan to take action.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

Rule WLM021: Subsystem transaction service class assigned to resource group

Finding: A subsystem transaction service class was assigned to a resource group.

Impact: This finding has no known impact on the performance of your system. The finding is presented to alert you to the fact that assigning a subsystem transaction service class to a resource group has no effect.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

Please refer to Section 4 for discussion of the relationship between server service classes and transaction service classes.

Of particular importance to this finding is the fact that the System Resources Manager (SRM) **does not accumulate resources** for subsystem transaction service classes. These transaction service classes are simply logical groupings of transactions based on the workload classification scheme, and the actual resources are used by (and attributed to) the address spaces processing the transactions.

For example, CICS transactions may be grouped into a transaction service class. CICS/ESA Version 4 reports information about the transactions to the Workload Manager (transaction response time and transaction delays). The CICS region is the address space which executes CICS tasks to process the CICS transactions, and the region address space uses resources.

Since subsystem transaction service classes do not consist of address spaces, assigning these service classes to a resource group has no effect. The Workload Manager can neither guarantee a minimum CPU service nor restrict the CPU service. The transactions do not use CPU service; the server service classes contain the address spaces using CPU service to process the transactions.

©Copyright 1995, Computer Management Sciences, Inc.

CPExpert produces Rule WLM021 when it detects that a service class describing subsystem transactions (e.g., CICS or IMS transactions) has been assigned to a resource group.

The following example illustrates the output from Rule WLM021:

```
RULE WLM021:  SUBSYSTEM TRANSACTIONS ASSIGNED TO A RESOURCE GROUP

  The IMS Service Class was assigned to the IMSRESGP Resource Group.
  This service class applied to subsystem transactions.  The Workload
  Manager does not apply resource group limits to subsystem transactions.
```

Suggestion: CPExpert suggests that you revise the service definition to eliminate the assignment of the resource group to the subsystem transaction service class. The Workload Manager does not apply resource group limits to service classes defined for subsystem transactions.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 7: Defining Resource Groups
OS/390 (V1R1):	Chapter 7: Defining Resource Groups
OS/390 (V1R2):	Chapter 7: Defining Resource Groups
OS/390 (V1R3):	Chapter 7: Defining Resource Groups
OS/390 (V2R4):	Chapter 7: Defining Resource Groups
OS/390 (V2R5):	Chapter 7: Defining Resource Groups
OS/390 (V2R6):	Chapter 7: Defining Resource Groups
OS/390 (V2R7):	Chapter 7: Defining Resource Groups
OS/390 (V2R8):	Chapter 7: Defining Resource Groups
OS/390 (V2R9):	Chapter 7: Defining Resource Groups
OS/390 (V2R10):	Chapter 7: Defining Resource Groups
z/OS (V1R1):	Chapter 7: Defining Resource Groups
z/OS (V1R2):	Chapter 7: Defining Resource Groups
z/OS (V1R3):	Chapter 7: Defining Resource Groups
z/OS (V1R4):	Chapter 7: Defining Resource Groups

Rule WLM022: Execution velocity goal specified for subsystem transaction service class

Finding: An execution velocity goal has been specified for a subsystem transaction service class.

Impact: This finding has a HIGH impact on the performance of the subsystem transaction service class.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

These transaction service classes are simply logical groupings of transactions based on the workload classification scheme, and the actual resources are used by (and attributed to) the address spaces processing the transactions.

For example, CICS transactions may be grouped into a transaction service class. CICS/ESA Version 4 reports information about the transactions to the Workload Manager (transaction response time and transaction delays). The CICS region is the address space which executes CICS tasks to process the CICS transactions, and the region address space uses resources.

Please refer to Section 4 for discussion of the relationship between server service classes and transaction service classes.

Of particular importance to this finding is the fact that the subsystem transaction service classes are not address spaces. Since they are not address spaces, the Workload Manager does not acquire CPU Using, CPU Delay, or Processor Storage Delay information for the subsystem transaction service classes. Consequently, the Workload Manager does not compute execution velocities for subsystem transaction service classes.

With OS/390 V2R10, IBM introduced an “exemption from transaction response time management” option. This option is available with APAR

©Copyright 1995, Computer Management Sciences, Inc.

OW43812 installed. With the APAR applied, organizations can specify whether an address space (CICS region or IMS region) will be managed based on the goals of the transactions that the region is serving, or managed based on the goals specified for the region itself. This option is exercised by using the new “Manage Region Using Goals Of:” field on the WLM ISPF “Modify Rules for the Subsystem Type” panel.

When “TRANSACTION” is entered in the “Manage Region Using Goals OF:” field, the region will be managed as a CICS/IMS transaction server by the WLM. “TRANSACTION” is the default specification. If “REGION” is entered in this field, the region will be managed based on the performance goal specified for the service class to which the region is assigned. This performance goal normally would be an execution velocity goal.

When “REGION” is specified, the WLM does not consider the region to be a “server” of transactions. Rather, the WLM server topology algorithms ignore the region when establishing server topology. Consequently, the goals for any transaction processed by the region will not be considered by the WLM when it determines whether service class periods meet goals and whether policy adjustment is necessary.

This consequence might have undesired implications if you specify goals for CICS or IMS transactions and some or all of those transactions are processed by a CICS or IMS region that has “REGION” specified in the “Manage Region Using Goals Of:” field. In this case, **performance of the transaction service class will not be considered when adjusting resource policy for the region**. This could have the undesired result of transactions not achieving the performance that you desire, simply because the transactions were processed by a CICS or IMS region that was managed based on the goals specified for the region. Alternatively, some transactions might receive better performance than desired because of the same “region-oriented” management by the WLM.

CPExpert produces Rule WLM022 when it detects that an execution velocity goal has been specified for a service class describing subsystem transactions (e.g., CICS or IMS transactions). Since the Workload Manager does not compute execution velocities for subsystem transaction service classes, it is illogical to specify an execution velocity goal for a subsystem transaction service class.

The following example illustrates the output from Rule WLM022:

```
RULE WLM022:  EXECUTION VELOCITY GOAL DEFINED FOR SUBSYSTEM TRANSACTIONS

  The CICSPERS Service Class was specified with an execution velocity goal
  of 50.  This service class applied to subsystem transactions.  The
  Workload Manager does not compute execution velocities for subsystem
  transaction service classes.  You should specify a response goal for
  this service class.
```

Suggestion: CPExpert suggests that you change the execution velocity goal to a response goal.

Alternatively, if you are running OS/390 Version 2 Release 10 (with APAR OW43812 installed) you might have deliberately assigned the "served" transaction service class to the region, and specified that the region be managed based on the goals specified for the region (normally, this would be an execution velocity goal). If this is the situation, you should consider the following alternatives:

- You can simply ignore this finding, but allow CPExpert to continue to check for such situations. The reason that you might wish to allow CPExpert continue to invoke Rule WLM022 is that your workload classification might change, new transactions might be added, CICS or IMS might route transactions to the CICS region or message processing region, etc. You might not be aware of the implications of the WLM assigning a specific transaction to a region managed by the region's goal.

- You can "turn off" Rule WLM022 as described in Section 2 of this WLM Component User Manual. This action should be taken if you become annoyed by Rule WLM022 being produced when you do not plan to take action.

Reference: MVS Planning: Workload Management

OS/390 (V2R4):	Chapter 8.3: Using Performance Periods
OS/390 (V2R5):	Chapter 8.3: Using Performance Periods
OS/390 (V2R6):	Chapter 8.3: Using Performance Periods
OS/390 (V2R7):	Chapter 8.3: Using Performance Periods
OS/390 (V2R8):	Chapter 8.3: Using Performance Periods
OS/390 (V2R9):	Chapter 8.3: Using Performance Periods
OS/390 (V2R10):	Chapter 8.3: Using Performance Periods
z/OS (V1R1):	Chapter 8.3: Using Performance Periods
z/OS (V1R2):	Chapter 8.3: Using Performance Periods
z/OS (V1R3):	Chapter 8.3: Using Performance Periods
z/OS (V1R4):	Chapter 8.3: Using Performance Periods

Rule WLM023: Too many service class periods may have been specified

Finding: CPExpert believes that too many service class periods may have been specified in the service definition.

Impact: This finding has a HIGH impact on the performance of service class periods.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: Users assign workload to a service class and specify the performance goal and goal importance of the service class. A service class can represent any collection of workload that can be classified using the workload classification schemes available with the Workload Manager. Each service class has Period 1 automatically defined. Optionally, users also may define multiple performance periods for service classes.

Service classes are much like performance groups available with MVS prior to Goal Mode, in that a workload classification scheme is used assign workloads to both service classes and performance groups. Another (unfortunate) similarity is that up to 999 performance groups can be defined and up to 1000 service classes can be defined. One result of these similarities is that users may perform a simple conversion of pre-Goal Mode performance group periods to Goal Mode service class periods.

This straightforward conversion is not a sound policy.

There was little performance impact of defining a large number of performance groups with pre-Goal Mode. There can be a **significant** performance impact of defining a large number of service class periods!

- The Workload Manager will adjust system resource allocation in an attempt to improve performance of only one service class period during

a policy adjustment interval[1]. Adjustment to improve performance of only one service class period is done because the Workload Manager must observe the results of the adjustment; whether the adjustment helped performance, hurt performance, or had no effect. If adjustments were made to improve performance of more than one service class period, it would be impossible to determine which adjustment helped or hurt which service class period. Consequently, actions to improve the performance of only one service class period is done during each policy adjustment interval.

The policy adjustment interval is 10 seconds. If too many service class periods have been defined, the Workload Manager may be able to adjust system resource allocation to help only a few service class periods (the most important service class periods with the worst performance). Performance of other service class periods may never be improved, or performance improvement actions may take a long time - simply because of the elapsed time necessary for the Workload Manager to make changes, collect data, analyze the effect of the changes, make additional changes, collect more data, and continue the process.

- Perhaps of equal significance is the overhead associated with analyzing service class periods. The Workload Manager tries to improve performance of the service class period with the worst performance at the highest goal importance. Resources may be taken from the least important service class period with the best performance. The Workload Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

  The Workload Manager will not add resources unless there is an appreciable net gain to the service class period receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class period unless the net gain to the receiver outweighs the net loss to the service class period the resources are being removed from. The overhead involved with the analysis and decision process increases as the number of service class periods becomes large.

IBM SRM/WLM developers have indicated that a small number of service class periods is desirable. They have observed that the Workload Manager algorithms typically become increasingly ineffective as the number of service class periods grows. As "rule of thumb" guidance, the developers have stated that most users should use only 25 service class

[1]The Workload Manager will consider adjustments to improve performance of several service classes (starting with the most important service class which has the highest Performance Index). If performance of the first service class analyzed cannot be appreciably improved, the Workload Manager will select the next worst performing service class, etc. After the Workload Manager has "committed" to a policy adjustment for a service class, it will stop analysis and adjust resources for no other service class.

periods and that no more than 30 service class periods normally should be defined.

From a practical matter, service class periods with a discretionary goal do not cause concern. This is because the Workload Manager groups all service class periods with a discretionary goal (and not assigned to a resource group) into the **$SRMDI00** internal service class, and treats them as a single service class.

CPExpert produces Rule WLM023 when it detects that (1) at least one service class period did not achieve its performance goal and (2) more than 30 service class periods with non-discretionary goals were defined.

The following example illustrates the output from Rule WLM023:

```
RULE WLM023:  TOO MANY SERVICE CLASS PERIODS MAY HAVE BEEN DEFINED

  CPExpert believes that you may have defined too many service class
  periods.  At least one service class did not meet its service goal,
  and a relatively large number of service class periods were defined.
  If too many service class periods are defined, the Workload Manager
  may not be able to adjust its resource allocation policies to meet
  the service goal of every service class period.  You specified a total
  of 41 service class periods with non-discretionary service goals.
```

Suggestion: CPExpert suggests that you revise your Service Policy to reduce the number of service class periods.

Reference: MVS Planning: Workload Management

OS/390 (V2R4):	Chapter 8.3: Using Performance Periods
OS/390 (V2R5):	Chapter 8.3: Using Performance Periods
OS/390 (V2R6):	Chapter 8.3: Using Performance Periods
OS/390 (V2R7):	Chapter 8.3: Using Performance Periods
OS/390 (V2R8):	Chapter 8.3: Using Performance Periods
OS/390 (V2R9):	Chapter 8.3: Using Performance Periods
OS/390 (V2R10):	Chapter 8.3: Using Performance Periods
z/OS (V1R1):	Chapter 8.3: Using Performance Periods
z/OS (V1R2):	Chapter 8.3: Using Performance Periods
z/OS (V1R3):	Chapter 8.3: Using Performance Periods
z/OS (V1R4):	Chapter 8.3: Using Performance Periods

"Effective use of MVS Workload Manager Controls", Berkel, Ed and Enrico, Peter (IBM Corporation), *CMG'95 Proceedings*, page 14.

Rule WLM024: More than three periods specified for a service class

Finding: More than three periods were specified for a service class.

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the service class periods involved. Under some circumstances, the finding could have a HIGH IMPACT on the performance of the service class periods involved. Additionally, there is an impact as a result of increased (and unnecessary) overhead.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: A service class may be broken into multiple service class periods. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional service class periods (although typically no more than one or two additional service class periods are defined). Each service class period can have its own performance goals, defined to the Workload Manager via the Workload Manager ISPF panels.

An address space (TSO transaction, batch job step, etc.) begins in Period 1 of the service class to which it is assigned. The address space transitions from Period 1 to Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction. The "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources. Performance period duration (DUR) values can be established so the system dynamically detects non-interactive (batch-like) transactions and migrates these transactions to lower performance periods, with less stringent performance goals and perhaps less goal importance.

The normal purpose of defining multiple service class periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are not conflicting with those address spaces requiring substantial resources.

It is instructive to examine the resources that may be in conflict. The resources mainly consist of CPU cycles, processor storage, and I/O operations.

- Prior to MVS/ESA SP5, the CPU resource was the main area of conflict. CPU dispatching priority is established at the performance group period

level. An address space in a particular performance group period executed at the same CPU dispatching priority as all other address spaces in the performance group period[1]. Non-interactive transactions could cause interactive transactions to be denied access to a processor for an unacceptable interval.

- Conflict with processor storage occurs when non-interactive transactions require either central storage or expanded storage, causing central storage page fault or expanded storage delays for interactive transactions. Non-interactive transactions often (but not always) require a significant amount of processor storage.

- Non-interactive transactions can cause conflict with I/O operations by either path, controller, or device delays to interactive transactions. Fortunately, (1) modern systems normally have multiple paths to devices and path delay is not usually a serious concern, (2) and non-interactive transactions normally do not access the same devices as interactive transactions, and (3) interactive transactions normally do not execute many I/O operations. The combined effect is that conflict for I/O operations rarely is a cause of serious performance problems between interactive and non-interactive transactions.

As the above discussion illustrates, the main concern prior to MVS/ESA SP5 was CPU dispatching. This concern has been almost completely eliminated with SP5.

- Prior to SP5, CPU dispatching was on a "first come first served" or FIFO basis from the Dispatcher's True Ready Queue. A non-interactive transaction could easily deny an interactive transaction from access to a processor for a prolonged period; the non-interactive transaction could retain control of a processor until it either completed or it voluntarily entered a Wait state.

- With SP5, the Dispatcher algorithm has been redesigned. One aspect of the redesign is that each interrupted dispatchable unit (a TCB or SRB) is placed at the end of the dispatching queue for its current dispatching priority.

  For example, suppose that a dispatchable unit was executing with a priority of 240. When the dispatchable unit was interrupted, it would be placed behind all other dispatchable units on the dispatch queue with a dispatching priority of 240 (but ahead of all dispatchable units with a dispatching priority of 239 and lower).

[1]This discussion ignores the implications of time-slicing algorithms and ignores the implications of the Mean-time-to-Wait (MTTW) dispatching algorithms. These special case algorithms are not normally used for Period 1 or Period 2 of multi-period performance groups.

Since interrupts occur frequently, there is a constant adjustment of dispatching order. This readjustment requires practically no additional overhead, as only queue pointers are being manipulated.

One effect of the redesign is to prevent any dispatchable unit from "seizing" a processor and denying access to other dispatchable units at the same priority. Since a dispatchable unit cannot deny access to other dispatchable units, a non-interactive transaction cannot deny CPU access to a Ready interactive transaction.

Consequently, with SP5, there is less need to "separate" non-interactive and interactive transactions, and there is less need to specify multiple periods. Nonetheless, the need to separate interactive and non-interactive transactions still exists. The SRM will be able to differentiate between interactive and non-interactive transactions only if the values specified for the DUR keyword roughly correspond to the resource requirements of trivial, interactive, and non-interactive transactions.

However, there can be a **significant** performance impact of defining an excess number of service class periods!

- The Workload Manager will adjust system resource allocation in an attempt to improve performance of only one service class period during a policy adjustment interval[2].

  Adjustment to improve performance of only one service class period is done because the Workload Manager must observe the results of the adjustment: whether the adjustment helped performance, hurt performance, or had no effect. If adjustments were made to improve performance of more than one service class period, it would be impossible to determine which adjustment helped or hurt which service class period. Consequently, the Workload Manager takes actions to improve the performance of only one service class period during each policy adjustment interval.

  The policy adjustment interval is 10 seconds. If too many service class periods have been defined, the Workload Manager may be able to adjust system resource allocation to help only a few service class periods (the most important service class periods with the worst performance). Performance of other service class periods may never be improved, or performance improvement actions may take a long time - simply because of the elapsed time necessary for the Workload Manager to make

[2]The Workload Manager will consider adjustments to improve performance of several service classes (starting with the most important service class which has the highest Performance Index). If performance of the first service class analyzed cannot be appreciably improved, the Workload Manager will select the next worst performing service class, etc. After the Workload Manager has "committed" to a policy adjustment for a service class, it will stop analysis and adjust resources for no other service class.

changes, collect data, analyze the effect of the changes, make additional changes, collect more data, and continue the process.

- Perhaps of equal significance is the overhead associated with analyzing service class periods. The Workload Manager tries to improve performance of the service class period with the worst performance at the highest goal importance. Resources may be taken from the least important service class period with the best performance. The Workload Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

  The Workload Manager will not add resources unless there is an appreciable net gain to the service class period receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class period unless the net gain to the receiver outweighs the net loss to the service class period the resources are being removed from. The overhead involved with the analysis and decision process increases as the number of service class periods becomes large.

Additionally, the system will incur other overhead caused by excess service class periods. As examples of the overhead:

- Additional SRM control blocks are created and processed.

- Additional Workload Manager control blocks are created and processed.

- RMF requires additional processing of the control blocks.

- SMF Type 72(Subtype 3) records are written for each service class period defined (regardless of whether the service class periods are used by the Workload Manager).

CPExpert produces Rule WLM024 when more than three periods are defined for a service class.

The following example illustrates the output from Rule WLM024:

```
RULE WLM024:  MORE THAN THREE PERIODS WERE SPECIFIED FOR A SERVICE CLASS

  More than three periods were specified for the TSOPROD Service Class.
  CPExpert believes that you may wish to revise the service class period
  structure to specify no more than three periods for the service class.
  Please refer to Rule WLM024 in the WLM Component User Manual for a
  discussion of the reasons for this recommendation.
```

Suggestion: CPExpert suggests that you review the performance period structure associated with the service class identified by this rule. Unless there are unusual circumstances, CPExpert suggests that you define no more than three[3] performance periods.

Some performance analysts have suggested that no more than two periods be defined[4]. Specifically, some authors have suggested defining TSO Period 1 with a very high duration (e.g., 50,000 service units) and defining TSO Period 2 to handle all other transactions. CPExpert strongly disagrees with this advice.

- Management at most installations would want a TSO transaction with 50,000 service units to be classified down with the "bottom feeders" (if for no other reason to discourage TSO users from sending in such a "transaction" rather than doing a batch job submit).

- The new dispatcher algorithms should minimize the CPU effect that a very large (e.g., 50,000 service unit) "transaction" would have on interactive TSO transactions in the same service class period. This would be true even if a lot of these lengthy transactions were using CPU cycles in Period 1. **That CPU time must come from somewhere, though**. All address spaces in a service class period have the same dispatching priority. The large transactions would be at the same CPU dispatching priority as TSO trivial and they would be using CPU time which probably should be available to other more important work.

- In addition to the effect on other workloads, since a very large (e.g., 50,000 service unit) "transaction" would be in the same service class period as interactive transactions, the large "transaction" would receive the "favored status" bestowed by the Workload Manager on service class periods with a short response goal. The Workload Manager gives the same expanded storage policy to all address spaces in a service class period with less than or equal to 20 seconds response goal.

[3]In fact, we would greatly appreciate being informed of any unusual circumstance in which more than three performance periods are required!

[4]Reference IBM TalkLink "Initial Dispatching" thread.

- Additionally, the Workload Manager gives a protective processor storage target to all address spaces in a service class period with a short response goal. In many shops, this favored status for large "transactions" could cause some serious performance problems[5]. If the large transactions were moved down to a period with over 20 seconds goal, the Workload Manager would treat them as individual address spaces, rather than collectively with all address spaces in the service class period.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

MVS/ESA 5.1.0 Presentation Guide (GG24-4137)
Section 4: Dispatcher Redesign

[5]A very large "transaction" is not a typical TSO interaction - who knows what evil would be generated by giving these "transactions" protective processor storage?

Rule WLM025: The service class period was inactive

Finding: The service class period was inactive for a significant percent of the RMF measurement intervals being analyzed.

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the Workload Manager and its ability to manage system resources to meet your performance goals.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The Workload Manager generates control blocks to describe each service class period defined in the service policy. These control blocks describe the performance goal, the importance of the goal, the resource group (if any) to which the service class is assigned, etc.

Additionally, the Workload Manager generates control blocks for each service class period to describe the resources used by the service class period, delays to the service class period, etc.

The Workload Manager periodically examines the SRM control blocks describing each address space and acquires samples[1] describing the state of each dispatchable unit of an address space (that is, each TCB or SRB associated with the address space). The Workload Manager accumulates the samples into counters which describe the state of the address space[2]. The samples are summarized by service class period.

The Workload Manager uses the samples in two basic ways: (1) the samples of CPU using and delays are used in computing the execution velocity of service class periods[3], and (2) the samples in various categories are used to guide the actions that the Workload Manager might take as it attempts to reallocate resources to meet specified performance goals.

---

[1]With MVS/ESA SP5.1 Goal Mode, the sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

[2]Address spaces can have more than one dispatchable unit (that is, more than one TCB or SRB). If an address space did have more than one dispatchable unit, the Workload Manager would accumulate state samples to describe the state of each dispatchable unit. To simplify the discussion, we shall refer to the sample states as being the "state" of the address space, with readers understanding that multiple dispatchable units could simultaneously be waiting on multiple events, or could be using the processor. Thus, for any single sampling, an address space could be counted in more than one state.

[3]Beginning with OS/390 (Release 3), the Workload Manager calculates execution velocity based on Total Using Samples divided by (Total Using Samples plus Total Delay Samples).

©Copyright 1997, Computer Management Sciences, Inc.

The underlying concept of a sampling process and the analysis of the samples obtained requires that a fairly large number of samples be taken. Basically, sampling normally requires sufficient samples such that the samples can be viewed as representative of the "population" being sampled.

If few address spaces or dispatchable units (TCBs, SRBs, or enclaves) are assigned to a service class, there will be few samples taken. Additionally, if those address spaces or dispatchable units that **are** assigned are primarily idle, few *meaningful* samples will be taken.

Please refer to Section 4 (Chapter 3) for additional information regarding the sampling process, the variables sampled, and how the Workload Manager uses the samples.

CPExpert examines each SMF Type 72 (Goal) record to determine whether sufficient samples are available for analysis. CPExpert simply compares the number of meaningful samples with the **MINSAMP** guidance variable contained in USOURCE(WLMGUIDE). CPExpert retains information for any service class period, by RMF interval, in which the number of meaningful samples is less than the MINSAMP guidance variable. After all RMF records have been analyzed, CPExpert determines whether any service class period was inactive (that is, whether few meaningful samples were available for analysis).

CPExpert produces Rule WLM025 when any service class period was inactive for more than 75% of the RMF intervals, and when this condition was true for all systems in the sysplex being analyzed. CPExpert ignores service class periods with discretionary goals, and ignores system service classes.

The following example illustrates the output from Rule WLM025:

```
RULE WLM025:  THE SERVICE CLASS PERIOD MAY BE UNNECESSARY

  STC_LOW (Period 1): This service class period was inactive for over
  75% of the RMF measurement intervals being analyzed (inactive means
  that less than 100 using and delay samples were collected by the SRM
  for the service class period during the RMF interval). Since the service
  class period was mostly inactive, you should re-evaluate whether this
  service class period is necessary.  Defining unnecessary service class
  periods (1) causes more overhead by the Workload Manager and (2) may
  cause the Workload Manager to be less responsive in managing system
  resources to meet your performance goals.  Please refer to Rule WLM025
  in the WLM Component User Manual for a discussion of the reasons for
  this recommendation.
```

Suggestion: If only a few meaningful samples are taken related to a service class period, the question arises as to why the service class period is in the service definition. There can be a **significant** performance impact of defining a large number of service class periods! Please consider the following:

- The Workload Manager makes its decisions mostly based on samples collected from service class periods executing in the sysplex. If few samples are taken for a particular service class period, the Workload Manager cannot validly make decisions regarding the service class period.

- The Workload Manager will adjust system resource allocation in an attempt to improve performance of only one service class period during a policy adjustment interval[4]. Adjustment to improve performance of only one service class period is done because the Workload Manager must observe the results of the adjustment: whether the adjustment helped performance, hurt performance, or had no effect. If adjustments were made to improve performance of more than one service class period, it would be impossible to determine which adjustment helped or hurt which service class period. Consequently, actions to improve the performance of only one service class period is done during each policy adjustment interval.

  The policy adjustment interval is 10 seconds. If too many service class periods have been defined, the Workload Manager may never be able to adjust system resource allocation to help more than a few service class periods (the most important service class periods with the worst performance). Performance of other service class periods may never be improved, or performance improvement actions may take a long time - simply because of the elapsed time necessary for the Workload Manager to make changes, collect data, analyze the effect of the changes, make additional changes, collect more data, and continue the process.

- Perhaps of equal significance is the overhead associated with analyzing service class periods. The Workload Manager tries to improve performance of the service class period with the worst performance at the highest goal importance. Resources may be taken from the least important service class period with the best performance. The Workload

[4]The Workload Manager will consider adjustments to improve performance of several service classes (starting with the most important service class which has the highest Performance Index). If performance of the first service class analyzed cannot be appreciably improved, the Workload Manager will select the next worst performing service class at the highest importance, etc. After the Workload Manager has "committed" to a policy adjustment for a service class, it will stop analysis and adjust resources for no other service class.

©Copyright 1997, Computer Management Sciences, Inc.

Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

The Workload Manager will not add resources unless there is an appreciable net gain to the service class period receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class period unless the net gain to the receiver outweighs the net loss to the service class period the resources are being removed from. The overhead involved with the analysis and decision process increases as the number of service class periods becomes large.

- IBM SRM/WLM developers have indicated that a small number of service class periods is desirable. They have observed that the Workload Manager algorithms typically become increasingly ineffective as the number of service class periods grows. As "rule of thumb" guidance, the developers have stated that most users should define no more than 25-30 service class periods.

From a practical matter, service class periods with a discretionary goal do not cause concern. This is because the Workload Manager groups all service class periods with a discretionary goal (and not assigned to a resource group) into the **$SRMDI00** internal service class, and treats them as a single service class.

CPExpert suggests that you re-evaluate whether the service class period is necessary. Unless unusual circumstances exist, CPExpert suggests that the service class period be removed from the service definition, and the work units reassigned to an existing service class.

You may wish to assign the work units to a report class, if you are concerned about reporting information related to the work units.

Alternatively, you can "turn off" CPExpert's analysis of inactive service class periods (and suppress this rule completely) by specifying **%LET INACTIVE = N;** in USOURCE(WLMGUIDE).

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

Thanks: Thanks to Glenn Bowman (Wakefern Food Corporation, NJ) for suggesting this rule.

Rule WLM026: Highest importance service class period had few samples

Finding: A service class period at the highest performance goal importance had few Using or Delay samples for a significant percent of the RMF measurement intervals being analyzed.

Impact: This finding should be viewed as generally having a MEDIUM IMPACT or HIGH IMPACT on the performance of the Workload Manager and its ability to manage system resources to meet your performance goals.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The Workload Manager generates control blocks to describe each service class period defined in the service policy. These control blocks describe the performance goal, the importance of the goal, the resource group (if any) to which the service class is assigned, etc.

Additionally, the Workload Manager generates control blocks for each service class period to describe the resources used by the service class period, delays to the service class period, etc.

The Workload Manager periodically examines the SRM control blocks describing each address space and acquires samples[1] describing the state of each dispatchable unit of an address space (that is, each TCB or SRB associated with the address space). The Workload Manager accumulates the samples into counters which describe the state of the address space[2]. The samples are summarized by service class period.

The Workload Manager uses the samples in two basic ways: (1) the samples of CPU using and delays are used in computing the execution velocity of service class periods[3], and (2) the samples in various categories are used to guide the actions that the Workload Manager might take as it attempts to reallocate resources to meet specified performance goals.

---

[1]With MVS/ESA SP5.1 Goal Mode, the sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

[2]Address spaces can have more than one dispatchable unit (that is, more than one TCB or SRB). If an address space did have more than one dispatchable unit, the Workload Manager would accumulate state samples to describe the state of each dispatchable unit. To simplify the discussion, we shall refer to the sample states as being the "state" of the address space, with readers understanding that multiple dispatchable units could simultaneously be waiting on multiple events, or could be using the processor. Thus, for any single sampling, an address space could be counted in more than one state.

[3]Beginning with OS/390 (Release 3), the Workload Manager calculates execution velocity based on Total Using Samples divided by (Total Using Samples plus Total Delay Samples).

©Copyright 1997, Computer Management Sciences, Inc.

The underlying concept of a sampling process and the analysis of the samples obtained requires that a fairly large number of samples be taken. Basically, sampling normally requires sufficient samples such that the samples can be viewed as representative of the "population" being sampled.

If few address spaces or dispatchable units (TCBs, SRBs, or enclaves) are assigned to a service class, there will be few samples taken. Additionally, if those address spaces or dispatchable units that **are** assigned are primarily idle, few **meaningful** samples will be taken.

Please refer to Section 4 (Chapter 3) for additional information regarding the sampling process, the variables sampled, and how the Workload Manager uses the samples.

CPExpert examines each SMF Type 72 (Goal) record to determine whether sufficient samples are available for analysis. CPExpert simply compares the number of meaningful samples with the **MINSAMP** guidance variable contained in USOURCE(WLMGUIDE). CPExpert retains information for any service class period, by RMF interval, in which the number of meaningful samples is less than the MINSAMP guidance variable.

After all RMF records have been analyzed, CPExpert determines whether any service class period was inactive (that is, whether few meaningful samples were available for analysis). Either Rule WLM025 or Rule WLM026 may be produced based on this analysis.

CPExpert produces Rule WLM026 when a service class period at the *highest goal importance* was inactive for more than 75% of the RMF intervals, and when this condition was true for all systems in the sysplex being analyzed.

The following example illustrates the output from Rule WLM026:

```
RULE WLM026:  HIGHEST IMPORTANCE SERVICE CLASS PERIOD HAD FEW SAMPLES

   STC_HI (Period 1): This service class period was inactive for over
   75% of the RMF measurement intervals being analyzed (inactive means
   that less than 100 using and delay samples were collected by the SRM
   for the service class period during the RMF interval).  However, the
   performance goal service class period was given the highest importance.
   Unfortunately, CPExpert cannot determine the type of work actually
   assigned to the service class period.  If the work assigned is started
   tasks, perhaps the started tasks can be assigned to the SYSSTC service
   class.  Please refer to Rule WLM026 in the WLM Component User Manual
   for a discussion of the reasons for this recommendation.
```

Suggestion: If only a few meaningful samples are taken related to a service class period with the highest importance, you may wish to consider the following alternatives:

- If the work assigned to the service class period identified by this rule consists of started tasks, you may wish to assign the started tasks to the **SYSSTC** service class. The SYSSTC service class automatically receives a high dispatching priority, and the dispatching priority is above the range of dispatching priorities that are dynamically managed by the Workload Manager[4].

  Since you have indicated that the service class is the *highest* importance, and since few meaningful samples were available for the Workload Manager to analyze, you may find that better performance is achieved if the work is assigned to the SYSSTC service class.

- If you cannot assign the work to the SYSSTC service class, you may wish to combine the work with other important work, and thus both reduce the number of service classes and provide the Workload Manager with more meaningful samples upon which to make its decisions:

  - The Workload Manager makes its decisions mostly based on samples collected from service class periods executing in the sysplex. If few samples are taken for a particular service class period, the Workload Manager cannot validly make decisions regarding the service class period.

  - The Workload Manager will adjust system resource allocation in an attempt to improve performance of only one service class period during a policy adjustment interval[5].

    Adjustment to improve performance of only one service class period is done because the Workload Manager must observe the results of the adjustment: whether the adjustment helped performance, hurt performance, or had no discernable effect.

    If adjustments were made to improve performance of more than one service class period, it would be impossible to determine which adjustment helped or hurt which service class period. Consequently,

[4]Address spaces in SYSSTC service class execute at dispatching priority FD (253) if APAR OW19265 is **not** applied, and execute at dispatching priority of FE (254) if OW19265 **is** applied.

[5]The Workload Manager will consider adjustments to improve performance of several service classes (starting with the most important service class which has the highest Performance Index). If performance of the first service class analyzed cannot be appreciably improved, the Workload Manager will select the next worst performing service class at the highest importance, etc. After the Workload Manager has "committed" to a policy adjustment for a service class, it will stop analysis and adjust resources for no other service class.

©Copyright 1997, Computer Management Sciences, Inc.

actions to improve the performance of only one service class period is done during each policy adjustment interval.

The policy adjustment interval is 10 seconds. If too many service class periods have been defined, the Workload Manager may never be able to adjust system resource allocation to help more than a few service class periods (the most important service class periods with the worst performance). Performance of other service class periods may never be improved, or performance improvement actions may take a long time - simply because of the elapsed time necessary for the Workload Manager to make changes, collect data, analyze the effect of the changes, make additional changes, collect more data, and continue the process.

- Perhaps of equal significance is the overhead associated with analyzing service class periods. The Workload Manager tries to improve performance of the service class period with the worst performance at the highest goal importance. Resources may be taken from the least important service class period with the best performance. The Workload Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

  The Workload Manager will not add resources unless there is an appreciable net gain to the service class period receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class period unless the net gain to the receiver outweighs the net loss to the service class period the resources are being removed from.

  The overhead involved with the analysis and decision process increases as the number of service class periods becomes large.

- IBM SRM/WLM developers have indicated that a small number of service class periods is desirable. They have observed that the Workload Manager algorithms typically become increasingly ineffective as the number of service class periods grows. As "rule of thumb" guidance, the developers have stated that most users should define only 25-30 service class periods.

You may wish to assign the work units to a report class, if you are concerned about reporting information related to the work units.

Reference: MVS Planning: Workload Management
MVS/ESA(SP 5): Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

Thanks: Thanks to Glenn Bowman (Wakefern Food Corporation, NJ) for suggesting this rule.

Rule WLM027: Service class periods have same velocity goal and importance

Finding: More than one service class period was defined with the same execution velocity goal and the same importance.

Impact: This finding should be viewed as generally having a MEDIUM IMPACT or HIGH IMPACT on the performance of the Workload Manager and its ability to manage system resources to meet your performance goals.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Users assign workload to a service class and specify the performance goal and goal importance of the service class. A service class can represent any collection of workload that can be classified using the workload classification schemes available with the Workload Manager. Each service class has Period 1 automatically defined. Optionally, users also may define multiple performance periods for service classes.

The Workload Manager tries to improve performance of the service class period with the worst performance at the highest goal importance. Resources may be taken from the least important service class period with the best performance. The Workload Manager will not simply remove and add resources; rather, the Workload Manager will analyze the net value of the planned action.

The Workload Manager will not add resources unless there is an appreciable net gain to the service class period receiving the resources. Within the same goal importance, the Workload Manager will not remove resources from a service class period unless the net gain to the receiver outweighs the net loss to the service class period the resources are being removed from. The overhead involved with the analysis and decision process increases as the number of service class periods becomes large.

IBM SRM/WLM developers have indicated that a small number of service class periods is desirable. They have observed that the Workload Manager algorithms typically become increasingly ineffective as the number of service class periods grows. Since the overhead increases with the number of service class periods, users should strive to minimize the service class periods defined.

CPExpert has noticed that than one service class period was defined with the same execution velocity goal and the same importance. This definition

might be useful if at least one of the service class periods were the second or subsequent period of a service class with multiple periods. However, the service classes identified by this finding had only one period.

If service classes have only one period and if they are defined with the same execution velocity goal and the same goal importance, they are treated identically by the Workload Manager. However, the service classes exist as distinct entities and thus create more overhead for the Workload Manager.

The following example illustrates the output from Rule WLM027:

```
RULE WLM027:  PERHAPS SERVICE CLASSES SHOULD BE COMBINED

CPExpert noticed that two or more service classes at the same Goal
Importance have been defined with an execution velocity goal, but
the actual values for the execution velocity goals were identical.  If
you have correctly specified the velocity goal and the Goal Importance,
CPExpert suggests that you combine the service classes into a single
service class.  Please refer to the WLM Component User Manual for a
discussion of this suggestion.

           SERVICE   EXECUTION           SERVICE   EXECUTION
           CLASS     VELOCITY            CLASS     VELOCITY
           APPCFEED     50               BATCHHI      50
```

Suggestion: CPExpert suggests that you review the service classes listed with Rule WLM027. If the execution velocity and goal importance for the service classes are consistent with your intent, you should consider combining the work associated with the service classes with identical execution velocity goals into a single service class.

If you wish to distinguish between the work in terms of resources or performance, you can define report classes to describe the work.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

"MVS Workload Manager Velocity Goals: What you don't know can hurt you", John Arwe, IBM Corporation, CMG'96 Proceedings.

Rule WLM030: Report class period is heterogeneous

Finding: At least one report class period was shown as "heterogeneous" by the SMF Type 72 information.

Impact: This finding should be viewed as NO IMPACT on the performance of your computer system. However, the workload information recorded by SMF may contain "double counting" of data. Consequently, the finding could have a HIGH IMPACT on accounting for the use of system resources, on billing for use of system resources, or on capacity planning efforts.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Two types of performance classes are reflected in data maintained by the Workload Manager: *service classes* and *report classes*.

- Service classes are used by the Workload Manager to determine whether performance goals are being met and to control the allocation of system resources to address spaces. A single transaction, job, Started Task, etc. can be controlled by only one service class (defined in the service policy).

- Report classes can be used to obtain information about the use of system resources at various levels of detail, based upon the specifications contained in the Workload Group definition of the service policy. Up to 999 report classes may be defined in a service policy. Earlier versions of MVS allowed the assignment of a workload element to up to four report performance groups. With Goal Mode, a single transaction, job, Started Task, etc. can be assigned to only **one** report class.

A report class period that is related to a subset of work in a **single** service class, is referred to as a **homogeneous** report class period. This term means that the report class period contains resource and performance information about work units in a **single** service class period, and that the performance goal and goal importance related to the report class period are obtained from a **single** service class period. As described below, with z/OS V1R2, it is possible to identify homogeneous report class periods.

A report class can be associated with work units that are assigned to multiple service class periods. That is, the workload classification scheme can associate work units to **multiple** service classes, while associating these work units to a **single** report class. When a report class is

©Copyright 1994, Computer Management Sciences, Inc.

associated to work units in multiple service classes, the report class is referred to as a **heterogeneous** report class. This term means that the report class period contains resource and performance information about work units in **multiple** service class periods.

Heterogeneous report classes can cause incorrect or misleading performance data, since the data collected is based on different goals, importance, or duration as specified for the multiple service class periods. As described below, with z/OS V1R2, it is possible to identify heterogeneous report class periods.

With z/OS V1R2, the heterogeneous report class will have a number of periods corresponding to the largest number of periods of any associated service class. For example, if Service Class 1 has two periods and Service Class 2 has three periods, and if work units assigned to Service Class 1 and Service Class 2 were assigned to Report Class 1, then Report Class 1 would have three periods (based on Service Class 2 having three periods).

With z/OS V1R2, SMF Type 72 (field R723CRS1: Report class period flags) contains information that describes whether a report class period is homogeneous or heterogeneous, on an RMF recording interval basis. A report class period is described as heterogeneous if more than one service class period was found contributing to the report class period in an RMF recording interval.

This new information means that it is possible for CPExpert to detect whether a report class period reverts between homogeneous and heterogeneous between RMF recording intervals, depending on whether work units actually executed in the corresponding service class periods during the RMF interval. Since the SMF information is reported on a system basis, it also is possible to determine whether a report class is homogeneous or heterogeneous on each system being analyzed by CPExpert.

Additionally, with z/OS V1R2, SMF Type 72 variable R723CLSC contains the name of the service class that last **contributed information** to the report class period. The performance goal and goal importance described in SMF Type 72 records for the report class period are obtained from the **last** service class period that contributed to the report class period. This new field can be used to associate the performance information to a specific service class period, which could be particularly useful if a report class period reverted between homogeneous and heterogeneous states from one RMF recording interval to the next.

While processing SMF Type 72 records, CPExpert extracts records for all report class periods in which R723CRS1 indicates that the report class

period is heterogeneous. CPExpert then summarizes these records by report class period and extracts information from all SMF Type 72 records that describe the report class periods, regardless of whether they are homogeneous or heterogeneous. The result is a compilation of information showing, by system, whether the report class period is homogeneous or heterogeneous for the entire SMF data being analyzed.

CPExpert produces Rule WLM030 to alert you that there is a problem with the performance data reflected by the report class period.

Recall that the SMF Type 72 records identify the last service class for each RMF interval that contributed to the report class period. Consequently, CPExpert lists the last service class that contributed to the report class period (obtained from the R723CLSC field). This identification of service classes associated with the report class period is not complete, since any particular service class might not be the “last” that contributed to the report class period. However, with a relatively large number of RMF intervals to analyze, it is likely that most associated service classes would be identified.

The following example illustrates the output from Rule WLM030:

```
RULE WLM030: REPORT CLASS PERIOD IS HETEROGENEOUS

   At least one report class period was shown as "heterogeneous" by
   the SMF Type 72 information.  A report class period is referred to
   as "heterogeneous" when the report class period is associated to work
   units in multiple service classes.  Heterogeneous report classes
   can cause incorrect or misleading performance data, since the data
   collected is based on different goals, importance, or duration as
   specified for the multiple service class periods.

   REPORT                                     LAST SERVICE CLASS
   CLASS       PERIOD      SYSTEM         CONTRIBUTING INFORMATION
   CICSCPSM      1          J90                   TESTSRVY
   CICSCPSM      1          JA0                   TESTSRVN
   CICSCPSM      1          JC0                   TESTSRVN
   CICSCPSM      1          JE0                   TESTSRVY
   CICSCPSM      1          JF0                   TESTSRVN
   CICSCPSM      1          JG0                   TESTSRVY
   CICSCPSM      1          JI0                   TESTSRVN
   CICSCPSM      1          TPN                   TESTSRVN
   CICSCPSM      1          Z0                    TESTSRVY
   CICSCPSM      1          JB0                   TESTSRVY
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- Unless you have unusual requirements, you should review the classification scheme of your report classes. The classification scheme should be revised to eliminate the assignment of work units from multiple service classes to the same report class.

- If you do have an unusual requirement, you should make sure that everyone who uses the heterogeneous report class period information is aware that heterogeneous report classes can cause incorrect or misleading performance data. This is because the data collected is based on different goals, importance, or duration as specified for the multiple service class periods.

- You can "turn off" CPExpert's analysis of heterogeneous report class periods by specifying **%LET WLM030 = OFF.**; in USOURCE(WLMGUIDE).

- You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is specified as **%LET CHCKPLCY=Y;**, CPExpert will not check the service definition for potential problems.

  Before you globally disable CPExpert's checking the service definition, you may wish to review other guidance variables. Many of the tests which CPExpert makes can be made inoperative by a guidance variable that applies to the specific test. The discussion of each finding describes the associated guidance variable.

Reference: MVS Planning: Workload Management
z/OS (V1R2): Chapter 10: Defining Report Classes

MVS Programming: Workload Management Services
z/OS (V1R2): Chapter 8.4.5 Interpreting Report Class Data

MVS Programming: Workload Management Services
z/OS (V1R3): Chapter 8.4.5 Interpreting Report Class Data

MVS Programming: Workload Management Services
z/OS (V1R4): Chapter 8.3.5 Interpreting Report Class Data

Rule WLM031: Dynamic Alias Management active but not I/O Priority

Finding: Dynamic alias management was active, but I/O priority management was not selected.

Impact: This finding should be viewed as LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depend on the amount of DASD I/O activity, the importance of the work requiring DASD I/O operations, and whether wait on DASD I/O caused performance problems for important work.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: With OS/390 Release 3, IBM introduced the I/O Priority Management option for Goal Mode. With I/O Priority Management, the system manages I/O priorities in the sysplex based on service class goals. The Workload Manager (WLM) dynamically adjusts the I/O priority based on how well each service class is meeting its goals and whether non-paging DASD I/O can contribute to meeting the goal. I/O priority queuing is used to control non-paging DASD I/O requests that are queued because the device is busy.

The I/O Priority Management process adjusts the I/O priority of non-paging DASD I/O operations (and includes the I/O activity (using and delay) as a part of the execution velocity calculation as described later in this document). However, these WLM algorithms operate only at the system level, and had no direct effect on I/O priority at the subsystem (controller and device) level until the Parallel Access Volumes (PAV) concept was available with OS/390 Release 9 (or with APAR OW39854 installed on OS/390 Release 7 or 8).

The IBM Enterprise Storage Server (ESS) supports concurrent or parallel data transfer operations to or from the same volume from the same system or system image. One feature of the PAV design is that I/O queuing is normally moved from queuing in the I/O Supervisor (commonly viewed as IOSQ time), to queuing in the 2105 ESS DASD subsystem. The 2105 ESS DASD subsystem can uses the I/O priority assigned to the I/O operation to prioritize the I/O access to the device.

With the PAV concept, *multiple device addresses* of a single control-unit image are *associated* with a single device (or parallel access volume).

©Copyright 1994, Computer Management Sciences, Inc.

With PAV, a base address is assigned to a CKD[1] logical device, and *alias device addresses* can be configured for the CKD logical device. While the base address is the actual unit address of a given volume, there can by many alias addresses assigned to a base address, and any or all of those alias addresses can be reassigned to a different base address.

These logical devices are identified by device numbers and represented by unit control blocks (UCBs) in the system. When an I/O is requested to a parallel access volume, the request identifies the base UCB and the system uses the base UCB or one of its alias UCBs, depending on their availability, to initiate the request. The Hardware Configuration Definition (HCD) is used to define multiple device numbers for a control unit that provides parallel access volumes.

When PAV was introduced, IBM also introduced optional *dynamic alias management*. With dynamic alias management, the Workload Manager can automatically perform the alias address reassignments to help work meet its goals and to minimize IOS queuing.

- If dynamic alias management is not enabled, the number of aliases to be assigned to each base device is determined by the parallel access volume's configuration application ("ESS Specialist") at IPL time or at the time a base device is varied online.

- If dynamic alias management is enabled, the Workload Manager monitors the device performance and automatically reassigns alias addresses from one base to another to help work meet its goals and to minimize IOS queuing.

The efficiency of the Workload Manger in assigning aliases to base devices depends on whether I/O Priority Management is enabled.

- If I/O Priority Management is enabled, and if the Workload Manager determines that a workload is not meeting its goal due to IOSQ time, then the Workload Manager attempts to find alias devices that can be moved to help that workload achieve its goal. Even if all work is meeting its goals, the Workload Manager will attempt to move aliases to the busiest devices to minimize overall I/O queuing.

- If I/O Priority management is **not** enabled, the Workload Manager will not use information about whether service classes meet goals and whether I/O queuing causes service classes to miss goals. Consequently, the Workload Manager will make decisions about assignment of alias

[1] CKD (count key data) is a term for a logical device that specifies the format of the logical data units on the device. The logical data unit is a track that can contain one or more records, each consisting of a count field, an optional key field, and an optional data field.

devices based only on an attempt to minimize overall IOS queuing, and these alias movement decisions will not take service class goals into consideration.

IBM strongly recommends that I/O Priority Management be specified if Dynamic Alias Management is specified.

CPExpert examines variable R723MSCF (Service/Report class flags) in SMF Type 72 information to detect whether Dynamic Alias Management and I/O Priority Management have been selected. CPExpert produces Rule WLM031 when Dynamic Alias Management has been selected, but I/O Priority Management was not selected.

The following example illustrates the output from Rule WLM031:

```
RULE WLM031: DYNAMIC ALIAS MANAGEMENT BUT NOT I/O PRIORITY MANAGEMENT

   Service Policy WLMPOL01, effective 17AUG2000:15:35:4, specified Dynamic
   Alias Management, but did not specify I/O Priority Management.  IBM
   strongly recommends that I/O Priority Management be specified if
   Dynamic Alias Management is specified.  If Dynamic Alias Management is
   specified but I/O Priority Management is not specified, the Workload
   Manager will not consider goals while managing Parallel Access Volumes.
   This means that the Workload Manager will make alias moves to minimize
   overall IOS queuing, but the Workload Manager will not consider service
   class goals when deciding the alias moves.
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- You should consider specifying I/O Priority Management so the Workload Manger will use information about goal achievement in its Dynamic Alias Management decisions.

  Additionally, I/O Priority Management is **required** with Channel Subsystem Priority Queuing available on z900 processors.

- There are, however, issues that might cause you to reconsider IBM's recommendation. These issues relate to how the Workload Manager creates and uses execution velocity.

With I/O Priority Management, samples of I/O activity[2] are used in the execution velocity formula. If I/O Priority Management is selected, non-paging DASD connect time and disconnect time are included in the "using" part of the execution velocity formula, and delays[3] waiting for non-paging DASD I/O operations are included in the "delay" part of the execution velocity formula.

There are two implications of the above algorithm:

- Execution velocity goals must be revised to include the performance index that would result once I/O activity is included in the execution velocity calculations. This revision requires some effort.

- The Workload Manager computes the execution velocity of a service class by applying the following algorithm:

$$\frac{\textit{using samples}}{\textit{using samples} + \textit{delay samples}} * 100$$

where:

using samples include:

- The number of samples of work using the processor (CPU Using).
- The number of calculated samples of work using non-paging DASD I/O resources (DASD connect state or DASD disconnect state). I/O using samples are included only if the installation has elected to include WLM-managed I/O.

delay samples include:

- The number of samples of work delayed for the processor (Denied CPU Delay or CPU Capping delay).
- The number of samples of work delayed for processor storage. Delay for processor storage includes:
  - Paging delay
  - Swap-in delay
  - Swapped out for multi programming (MPL) reasons

---

[2] The "I/O samples" are not actually samples, but are numeric values computed by multiplying the amount of measured I/O activity by the WLM sample period of four samples per second. The WLM samples activity (such as CPU activity) and delays (such as CPU delay or processor delay) for the execution velocity formula. Since the DASD I/O activity and delay are measurements made at the controller level, it is not possible for the WLM to sample the activity or delay state of the device. However, Don Deese pointed out to IBM SRM/WLM developers that simply multiplying the DASD activity and delay measured time by the WLM sample rate would yield calculated sample counts that would be equivalent to actual sample counts. These calculated sample counts are used by I/O Priority Management and execution velocity calculations.

[3] Non-paging DASD I/O delays include IOS queue delays, SubChannel pending delays, and control unit queue delays. Note that DASD disconnect time is not included in the execution velocity delay calculations, but is including in the "using" component of the calculation!

- Server address space creation delay
- Initiation delays for batch jobs in WLM-managed job classes

The result from the algorithm is multiplied by 100, to yield an execution velocity ranging from 0 (when the address space did not use the CPU) to 100 (when the address space was not delayed for any reason controlled by the SRM).

Many analysts were dismayed that the DASD disconnect time is included in the "using" part of execution velocity. IBM's rationale for including disconnect in the "using" part is that the Workload Manager can take action to reduce delays. Since the WLM cannot take action to decrease disconnect, so there is little point in including disconnect in the "delay" part of the execution velocity calculations.

Unfortunately, the effect of including disconnect time in the execution velocity calculation meant that execution velocities were not a reliable indicator of performance. As the DASD disconnect time increases, the DASD response time would increase (which meant that DASD performance was getting worse). However, the execution velocity algorithm would calculate that performance was improving with higher disconnect time, since disconnect was included in the "using" part of execution velocity.

Even worse, high disconnect time and resulting large execution velocity would cause the Performance Index to be small. The low Performance Index would make it appear the service class is performing better than if disconnect time were not counted. This could allow a service class to become a donor of a I/O resource, when the devices it was using were performing poorly.

Additionally, since the DASD disconnect time often would be large relative to other potential causes of using or delay, the disconnect time would dominate the execution velocity algorithm.

IBM responded to this problem with APAR OW47667, in which disconnect time is no longer counted as productive I/O time. Also, Disconnect time is not counted as I/O delay, because there is nothing the Workload Manager can do to reduce disconnect time.

The APAR states "This change will affect achieved velocities for systems which have the I/O priority queuing option set to ON in the WLM policy. Achieved velocities will be the same or lower, depending on how much disconnect time the service class experiences. Customers should review velocity goals in their WLM policy and adjust downward if needed."

©Copyright 1994, Computer Management Sciences, Inc.

Consequently, CPExpert recommends that you install APAR OW47667 before turning on I/O Priority Management.

- You can "turn off" Rule WLM031 by specifying **%LET WLM031 = OFF.**; in USOURCE(WLMGUIDE) if decide not to implement I/O Priority Management.

- You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is specified as **%LET CHCKPLCY=Y;**, CPExpert will not check the service definition for potential problems.

  Before you globally disable CPExpert's checking the service definition, you may wish to review other guidance variables. Many of the tests which CPExpert makes can be made inoperative by a guidance variable that applies to the specific test. The discussion of each finding describes the associated guidance variable.

Reference: MVS Planning: Workload Management

OS/390 (V2R9):	Chapter 11.3:	Specifying Dynamic Alias Management
OS/390 (V2R10):	Chapter 11.3:	Specifying Dynamic Alias Management
z/OS (V1R1):	Chapter 11.3:	Specifying Dynamic Alias Management
z/OS (V1R2):	Chapter 11.3:	Specifying Dynamic Alias Management
z/OS (V1R3):	Chapter 11.3:	Specifying Dynamic Alias Management
z/OS (V1R4):	Chapter 11.3:	Specifying Dynamic Alias Management

Rule WLM032: Server was assigned CPU protection but most work done in support of lower importance work

Finding: A server (CICS or IMS region) was assigned long-Term CPU Protection, but most CPU work in the CICS or IMS region was done in support of lower importance transaction service classes.

Impact: This finding should be viewed as MEDIUM IMPACT or HIGH IMPACT on the performance of your computer system. The level of impact depend on the amount of CPU activity that is given to lower importance work, the importance of the work that might be denied CPU service because CPU protection was given to work processed by the server, and whether management is concerned about mis-allocation of CPU resources.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: With OS/390 Release 10 (and with APAR OW43855 installed), IBM introduced the Long-Term CPU Protection option for Goal Mode. When Long-Term CPU Protection is assigned to a service class, the Workload Manager attempts to ensure that less important service class periods will have a lower dispatch priority than the service class that is assigned Long-Term CPU Protection.

The Long-Term CPU Protection option was implemented in OS/390 Release 10 - long after Goal Mode was announced. This option was released so late after Goal Mode was announced because CPU Protection significantly deteriorates the basic concepts of Goal Mode design.

One reason that IBM introduced Goal Mode as a way of minimizing the requirement that systems personnel have detailed knowledge of system-internals. Prior to Goal Mode, system programmers and performance analysts needed to understand the detailed internal logic of the System Resources Manager (SRM), so they could make specific changes to the IEAIPSxx and IEAOPTxx members of SYS1.PARMLIB to obtain good performance of their computer systems. Sadly, individuals with such detailed knowledge were scarce and organizations were not always able to obtain good performance because of the lack of knowledge.

With IBM's plans for increasingly complex environments (parallel sysplex, coupling facility, etc.), IBM realized that an increased level of knowledge would be required to optimize performance for these environments. IBM designed the Workload Manager (WLM) as a way to automate the

©Copyright 1994, Computer Management Sciences, Inc.

performance tuning process, and reduce the level of knowledge required to manage systems performance.

The basic concept of the Workload Manager is that installation personnel should:

- Stratify their work into several levels of importance,
- Set performance goals for work at each level of importance, and
- Let the system automatically allocate resources so the performance goals would be met and the optimum amount of work would be performed.

IBM SRM/WLM developers designed complex algorithms to ensure that goals were met for important work, and that system resources were distributed to all work to optimize throughput. In most instances, the algorithms work extremely well, if the goals are set to match management or user expectations, and the importance of work is correctly specified.

There are, however, a few design decisions that can cause performance problems for important work, or can cause management consternation.

- **Small CPU consumer**. With Goal Mode, the SRM/WLM developers introduced the "small CPU consumer" algorithm. The intent of the small CPU consumer algorithm was to identify those few tasks that (1) used a very small amount of CPU and (2) would quickly finish if given a modest amount of CPU time. The WLM attempted to identify these small CPU consumer tasks, given them a high priority access to a CPU, and expect that the tasks would quickly finish and be removed from the system. When Goal Mode was first introduced, the WLM gave these small CPU consumer tasks very high dispatching priority (the dispatching priority was 254; right below Master (with 255), and above all other work).

  Giving small CPU consumers such high CPU dispatching priority was not popular, since the CPU dispatching priority of unimportant batch work could be higher than the CPU dispatching priority of SYSSTC (system started tasks). Consequently, IBM changed the CPU dispatching priority scheme to be 255 (Master), 254 (SYSSTC), and 253 (small CPU consumers).

  The resulting scheme still was not popular, because this scheme meant that the CPU dispatching priority of small CPU consumer work would be higher than work with goals (work with goals is the production work that has performance goals and importance assigned). Sadly, the small CPU consumers often consisted of low-importance work (such as short batch jobs), and installation personnel would see unimportant work have a

higher CPU dispatching priority than the most important production work. Complicating the issue was that the WLM would not always recognized in a timely fashion that work that had been identified as “small CPU consumers” had, in fact, changed their processing characteristics and were no longer small CPU consumers. Thus, the “small CPU consumers” would have a negative effect on the performance of critical work with goals. For a very few users, this negative effect (even though it lasted for a very short time) would so severely effect their operation that they would go back to Compatibility Mode.

- **Allocation of CPU dispatching priority**. The WLM attempts to maximize throughput, while meeting performance goals. As a part of the resource allocation algorithms, the WLM can determine that a particular service class period is missing its goal because it is being denied access to a CPU. In this case, the WLM can examine the CPU use of service class periods with a higher dispatching priority to see whether those service class periods would be harmed if dispatching priorities were realigned. The WLM can decide that there would be no detrimental effect on the service class periods with a higher CPU dispatching priority, if the service class missing its goal were placed above the service class periods with a higher CPU dispatching priority. These algorithms are very conservative, in that they analyze history to make sure that the higher importance work was always exceeding its goal, that the higher importance work would not be denied CPU access and fail to meet its goal if the lower importance work had a higher CPU dispatching priority, and that assigning a higher CPU priority to the lower importance work would help the lower importance work meet its goal.

  The WLM concept is that a lower importance service class needs access to a CPU and that giving this access would not cause a higher importance service class service class to miss its goal. In most cases, this concept works well.

  There have been, however, situations in which the algorithms did not work well because one of the service class periods (either the low importance work or the high importance work) changed its processing characteristics. If the low importance work with a higher CPU dispatching priority suddenly began requiring more CPU, the WLM would take action to reduce its CPU dispatching priority. Unfortunately, the WLM requires some elapsed time to notice that the situation has changed and to take remedial action. During this elapsed time, performance could suffer for the high importance work.

- **Management consternation**. Regardless of whether work with a high importance met its goal, management often becomes concerned if they observe lower importance work having a CPU dispatching priority higher

than high importance work. Management often feels that there is something intuitively wrong with the situation, and do not approve.

For these reasons, some very important IBM users adamantly demanded that the WLM CPU dispatching priority allocation algorithms be changed. Consequently, IBM introduced the Long-Term CPU Protection option. As mentioned earlier, when Long-Term CPU Protection is assigned to a service class, the Workload Manager attempts to ensure that less important service class periods will have a lower dispatch priority than the service class that is assigned Long-Term CPU Protection.

In general, implementing Long-Term CPU Protection is not a good idea. This is because (except in very special situations) Long-Term CPU Protection automatically removes much of the flexibility in the WLM algorithms as they attempt to allocate resources so that all service class periods meet their performance goals, and as they attempt to maximize throughput. **Only if you have very time-critical work that cannot tolerate any occasional delay for CPU, should you implement Long-Term CPU Protection!**

That said, there is one additional aspect of Long-Term CPU Protection that must be understood: if Long-Term CPU Protection is implemented for a server (CICS or IMS region), the CPU protection applies to **all work** processed by the server, regardless of the importance of the associated transaction service classes.

It is common for a server to process transactions that have been assigned to more than one transaction service class, since some transactions are very important while other transactions are less important. The transactions can be classified and assigned to different transaction service classes, and the transaction service classes can have different performance goals and goal importance. If transactions in these different transaction service classes are processed by a server with CPU protection assigned, all transactions receive the same CPU protection, regardless of their importance.

CPExpert examines variable R723MSCF (Service/Report class flags) in SMF Type 72 information to detect whether Long-Term CPU Protection has been implemented for a service class. CPExpert further determines whether that particular service class is a server (CICS or IMS region) for more than one transaction service class and whether the transaction service classes have different goal importance. If this is the case, CPExpert sums the *service* given to transaction service classes.

The *service* given to a transaction service class by a server is extracted from the “Service Class Served Data Section” of SMF Type 72 records. The total service provided by the server to all transaction service classes

is summed, and a percent of service given to each transaction service class is calculated.

CPExpert separately sums the service given to transaction service classes between the service provided to the transaction service classes at the highest goal importance, and the service provided to all other transaction classes at a lower goal importance.

CPExpert produces Rule WLM032 when the total service provided to the transaction service classes at the highest goal importance is less than the value specified for the **PCTSERVC** guidance variable in USOURCE(WLMGUIDE). The default value for the PCTSERVC guidance variable is 50%, indicating that Rule WLM032 would be produced when the most important transaction service classes received less than 50% of the service provided by the server to all transaction service classes.

The following example illustrates the output from Rule WLM032:

```
RULE WLM032: SERVER PROTECTED BUT MOST WORK DONE FOR LOWER IMPORTANCE

   Service Policy WLMPOL01 (effective 12SEP2002:14:20:20) specified CPU
   Protection for the CICSRGN service class.  The CICSRGN service class
   was a server for transaction service classes that had different Goal
   Importance levels.  Some of the work done by the CICSRGN service class
   was in support of a high Goal Importance, but more than 50% of the work
   supported transaction service classes with a low Goal Importance.  The
   below information shows how often CICSRGN provided service to transaction
   service classes at different importance:

                                TRANSACTION        GOAL          PERCENT
   MEASUREMENT INTERVAL       SERVICE CLASS    IMPORTANCE       SERVICE
   14:45-15:00,03OCT2002        CICS                2              1.7
   14:45-15:00,03OCT2002        CICSDEFA            3             49.0
   14:45-15:00,03OCT2002        CICSLONG            3             47.3
   14:45-15:00,03OCT2002        CICSCONV            3              2.1
   14:45-15:00,03OCT2002        CICSMISC            3              0.0
```

Suggestion: Rule WLM032 alerts you to the potential resource allocation problem resulting from low importance work receiving most of the service provided by the server (CICS or IMS region). Since Long-Term CPU Protection was specified for the server service class, this low importance work is processed at a high CPU dispatching priority. Since the work is low importance, you should consider whether management wishes this low importance work to receive the favorable CPU protection. If Rule WLM032 is produced regularly, CPExpert suggests that you consider the following alternatives:

- Remove Long-Term CPU Protection from the CICS or IMS server service class. In most cases, this is the best alternative. As explained above, the Long-Term CPU Protection option is suited only for very special

cases. For the normal case, the WLM resource allocation algorithms will provide adequate CPU dispatching priority for work with a high importance. If that work should miss its performance goal, the WLM will adjust resources (including CPU dispatching priority) as necessary to ensure that the work achieves its performance goal.

- Revise the work classification scheme to remove the low importance work from the CICS or IMS server, and assign these transactions to a CICS or IMS server that does not have the Long-Term CPU Protection option enabled. This alternative might require that a separate CICS or IMS region be established, or might be possible using existing regions that serve other low importance transactions.

- You can alter CPExpert's analysis by modifying the PCTSERVC guidance variable in USOURCE(WLMGUIDE).

- You can "turn off" Rule WLM032 by specifying **%LET WLM032 = OFF.**; in USOURCE(WLMGUIDE) if decide not to implement I/O Priority Management.

- You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is specified as **%LET CHCKPLCY=Y;**, CPExpert will not check the service definition for potential problems.

  Before you globally disable CPExpert's checking the service definition, you may wish to review other guidance variables. Many of the tests which CPExpert makes can be made inoperative by a guidance variable that applies to the specific test. The discussion of each finding describes the associated guidance variable.

Reference: MVS Planning: Workload Management

OS/390 (V2R10):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R1):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R2):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R3):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R4):	Chapter 12.2: Long-Term CPU Protection

Rule WLM050: The number of available page slots is low

Finding: CPExpert has determined that the number of page slots available to the Auxiliary Storage Manager (ASM) has become so low that the ASM is danger of running out of slots.

Impact: This finding can have a MEDIUM impact or HIGH impact on the performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: The SRM checks for auxiliary storage shortage at two levels:

- The SRM detects when 70% of the local auxiliary storage slots are allocated. When less than 30% are available the SRM notifies the operator (using message IRA200I) that a shortage of auxiliary storage paging space exists. The SRM will reject LOGON, MOUNT, and START commands until the shortage of auxiliary storage is relieved.

- The SRM detects when 85% of the local auxiliary storage slots are allocated. When less than 15% are available the SRM notifies the operator (using message IRA201I) that a critical shortage of auxiliary storage paging space exists. The SRM will reject LOGON, MOUNT, and START commands until the shortage of auxiliary storage is relieved.

The SRM will attempt to identify address spaces with rapidly increasing auxiliary storage requirements, at both of the above levels. These address spaces will be swapped out (reason code Auxiliary Storage Shortage) until the shortage is relieved.

In order to help prevent an auxiliary storage shortage, CPExpert will produce Rule WLM050 when the available auxiliary storage is less than 50% of the total space allotted. CPExpert thus provides an "early warning" of a potential problem.

Suggestion: CPExpert suggests that additional slots be allocated to the local page data sets.

©Copyright 1994, Computer Management Sciences, Inc.

Reference: MVS Initialization and Tuning Reference, MVS/ESA SP5.1
Section 2.3.9.1.7 (Page Space Shortage discussion)

Please note that while this reference applies to MVS/ESA SP5 (Compatibility Mode), the finding is applicable to versions of MVS after SP5.

The two thresholds described above are controlled by the constants MCCASMT1 (for the low threshold) and MCCASMT2 (for the critical shortage) located in the SRM module IRARMCNS.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM051: The number of local page data sets may be inadequate

Finding: CPExpert has determined that the number of local page data sets does not always allow the Auxiliary Storage Manager to accomplish parallel swapping.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The level of impact depends upon how many physical swaps to auxiliary storage are done by the SRM and how much the average swap is delayed.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: Swaps to auxiliary storage occur in "swap sets" of 30 page frames if local page data sets are used. The number of pages to be swapped out is divided into these swap sets. For example, if 95 page frames are swapped, there would be 4 swap sets. Three of the swap sets would be 30 page frames and 1 swap set would be 5 frames.

The Auxiliary Storage Manager (ASM) can swap the swap sets concurrently to local page data sets if sufficient local page data sets exist. For example, if there were four swap sets and four local page data sets, the ASM would initiate concurrent swap operations to place a swap set out to each of the local page sets. This process is called "parallel swapping". Parallel swapping allows the actual swap time to be lowered by the number of local page data sets actually used.

Parallel swapping can have a significant reduction in swap-in delay time. This reduction can significantly improve the performance of the address space being swapped in.

Parallel swapping occurs only if swapping is done to auxiliary storage. Thus, if most swaps are logically swapped or are swapped to expanded storage, parallel swapping will have only a modest effect on swap delay time. However, parallel swapping can significantly improve performance when (1) central storage is constrained, (2) expanded storage is unavailable, and (3) swap-in causes a significant delay to a service class period.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert analyzes the page/swap configuration when a service class period missed its performance goal and swap-in was a major performance factor for the service class period.

CPExpert computes the average swap-out size by dividing the number of physical swaps into the number of swap-out pages (SMF71SOT). The average swap-out size is divided by 30 to yield the average number of swap sets. Rule WLM051 is produced if the average number of swap sets is greater than the number of local page data sets.

The following example illustrates the output from Rule WLM051:

```
RULE WLM051:  THE NUMBER OF LOCAL PAGE DATA SETS MAY BE INADEQUATE

   CPExpert has determined that the number of local page data sets does
   not always allow the Auxiliary Storage Manager to accomplish parallel
   swapping.  During the following intervals, at least one service class
   missed its performance objective, and a major cause of delay was waiting
   for pages or waiting for swap-in to complete.  Page delay or swap-in
   delay can be caused by inefficient parallel swapping.  In order to
   achieve parallel swapping, you should consider allocating more local
   page data sets.

                                 AVERAGE      LOCAL PAGE DATA     SUGGESTED
   MEASUREMENT INTERVAL        SWAP SIZE      SETS ALLOCATED    ALLOCATION
   14:00-14:15, 01MAR1994         111                2                4
   14:45-15:00, 01MAR1994          99                2                4
```

Suggestion: CPExpert suggests that you consider increasing the number of local page data sets so that parallel swapping can occur. Effective parallel swapping requires that local page data sets be placed onto dedicated packs (or packs with very low activity data sets). Additionally, these packs should be attached to paths which have low utilization.

Please note that CPExpert will check whether significant migration occurred. If significant migration occurred, CPExpert will produce Rule WLM059 if the number of local page data sets is insufficient to allow parallel I/O operations for migration of pages from expanded storage to auxiliary storage.

Rule WLM059 may suggest a different number of local page data sets than does Rule WLM051. The reason for this difference is that Rule WLM059 analyzes the number of local page data sets based on the number of pages migrated, while Rule WLM051 analyzes the number of local page data sets based on the average swap size. You should consider the **larger** of the two suggested values.

Rule WLM052: The number of page slots allocated may not be sufficient to allow the contiguous slot algorithm to be effective

Finding: CPExpert has determined that the number of page slots allocated to local page data sets may not always allow the Auxiliary Storage Manager to implement the *contiguous slot algorithm*.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The level of impact depends upon how many physical swaps to auxiliary storage are done by the SRM and how much the average swap is delayed.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: Swaps to auxiliary storage occur in "swap sets" of 30 page frames if local page data sets are used. The number of pages to be swapped out is divided into these swap sets. For example, if 95 page frames were swapped, there would be 4 swap sets. Three of the swap sets would be 30 page frames and 1 swap set would be 5 frames.

The Auxiliary Storage Manager (ASM) will attempt to attempt to write the swap set to a contiguous set of slots. The logic necessary to locate contiguous slots is called the *contiguous slot algorithm*.

If the contiguous slot algorithm is successful in locating contiguous slots, only one seek is required to write the swap set (and only one seek would be required to read the swap set when the address space is swapped back in). Additionally, there would be no latency between page writes (or page reads upon swap-in). The effect of the contiguous slot algorithm is to minimize the device service time per page and to minimize the device utilization.

The contiguous slot algorithm is effective only if there is a sufficiently large number of slots allocated to local page data sets so that fragmentation of the data sets does not preclude the ASM finding contiguous slots. As a general guidance, the number of slots allocated should be at least four times the number of slots used.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert analyzes the page/swap configuration when a service class period missed its performance goal and swap-in was a major performance factor for the service class period.

CPExpert analyzes the number of slots allocated to local page data sets, versus the number of slots in use. This analysis is based upon information contained in SMF Type 75 records (SMF75SLA and SMF75MXU, respectively).

Rule WLM052 is produced if the number of slots allocated for local page data sets is less than four times the number of slots used.

The following example illustrates the output from Rule WLM052:

```
RULE WLM052:  THE NUMBER OF ALLOCATED PAGE SLOTS MAY BE INSUFFICIENT

  The number of slots allocated for local page data sets may not always
  allow the Auxiliary Storage Manager to implement the contiguous slot
  algorithm.  During the measurement intervals shown below, the number of
  available slots was less than the desired percent of slots allocated,
  and at least one service class missed its performance goal because of
  paging delays or swap-in  from auxiliary storage.  As a general rule,
  the number of allocated slots should be about four times the average
  number of slots used to ensure that the contiguous slot algorithm has
  sufficient space.

                              AUXILIARY STORAGE      SLOTS       PERCENT
   MEASUREMENT INTERVAL        SLOTS ALLOCATED        USED      SLOTS USED
   7:30- 8:00, 16AUG1995            36,000          16,427        45.6
   8:00- 8:30, 16AUG1995            36,000          16,090        44.7
```

Suggestion: CPExpert suggests that you consider increasing the number of slots allocated to local page data sets so that the contiguous slot algorithm can be effective.

Reference: "Paging Analysis in an Expanded Storage Environment", Beretvas, Thomas (IBM Corporation), *CMG '87 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL.

Rule WLM053: The number of page slots allocated may not be sufficient for effective block paging

Finding: CPExpert has determined that the number of page slots allocated to local page data sets may not always allow the Auxiliary Storage Manager to implement the contiguous slot algorithm. **This finding relates to the block paging algorithms**.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The level of impact depends upon how many block pages are directed to auxiliary storage or are migrated to auxiliary storage.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: Prior to SP4.2, the Real Storage Manager (RSM) would steal a limited number of frames from any address space (the limit would be either 2 or 10 frames, depending upon system conditions). With SP4.2, the limit on the number of frames stolen from an address space has been removed.

The Page Steal Routine (IRARMPR5) in the System Resources Manager (SRM) sets the **target** number of frames to steal as a function of the number required to replenish the Available Frame Queue (that it, it takes as many frames as are required to reach the RCEAFCOK value). The RSM uses the target to steal pages based upon UIC. Thus, for a given UIC value, the RSM will steal up to the target number of frames from a single address space.

Additionally, the Page Trim Routine (IRARMPR9) in the SRM will trim pages from swapped out address spaces based upon the UIC.

The frames stolen or trimmed from an address space at a given UIC are moved to expanded storage or to auxiliary storage as a block, **if the pages are contiguous in virtual storage**. The RSM maintains information about the pages in a block, when the block is formed.

If the block is moved to auxiliary storage (either directly or migrated from expanded storage), the RSM requests that the contiguous slot algorithm be used. Thus, the Auxiliary Storage Manager will place the block into contiguous page slots on auxiliary storage if contiguous slots are available.

©Copyright 1994, Computer Management Sciences, Inc.

If the contiguous slot algorithm is successful in locating contiguous slots, only one seek is required to write the block. Additionally, there would be no latency between page writes. The effect of the contiguous slot algorithm is to minimize the device service time per page and to minimize the device utilization.

When a page fault subsequently causes a page in the block to be retrieved from auxiliary storage, all pages in the block **following the page referenced** will be brought into storage, in anticipation that these pages might also be referenced. The rationale for this is that there may be a high probability that other pages in the block will be referenced once any page in the block is referenced. Recall that a block consists of contiguous pages at the same UIC. If other pages at the same UIC are referenced, page fault resolution from auxiliary storage can be avoided if the remainder of the block is brought into central storage.

The contiguous slot algorithm is effective only if there is a sufficiently large number of slots allocated to local page data sets so that fragmentation of the data sets does not preclude the ASM finding contiguous slots. As a general guidance, the number of slots **allocated** should be at least four times the number of slots actually **used**.

CPExpert analyzes the number of slots allocated to local page data sets, versus the number of slots in use. This analysis is based upon information contained in SMF Type 75 records (SMF75SLA and SMF75MXU, respectively). Additionally, CPExpert verifies that blocked pages were written to auxiliary storage.

Rule WLM053 is produced if the number of slots allocated for local page data sets is not four times the number of slots used and if some service class missed its performance goal because of page-in or swap-in delays.

The performance benefits of block paging from auxiliary storage are defeated if the contiguous slot algorithm is not effective.

The following example illustrates the output from Rule WLM053:

```
RULE WLM053:  THE NUMBER OF ALLOCATED PAGE SLOTS MAY BE INSUFFICIENT

  The number of slots allocated for local page data sets may not always
  allow the Auxiliary Storage Manager to implement the contiguous slot
  algorithm for block paging. During the measurement intervals shown below,
  the number of available slots was less than the desired percent of slots
  allocated, and at least one service class missed its performance goal
  because of paging delays from auxiliary storage.  As a general rule, the
  number of allocated slots should be about 4 times the average number of
  slots used to ensure that the contiguous slot algorithm has sufficient
  space.

                              AUXILIARY STORAGE      SLOTS       PERCENT
   MEASUREMENT INTERVAL        SLOTS ALLOCATED        USED      SLOTS USED
   7:30- 8:00, 16AUG1995            36,000          16,427        45.6
   8:00- 8:30, 16AUG1995            36,000          16,090        44.7
```

Suggestion: CPExpert suggests that you consider increasing the number of slots allocated to local page data sets so that the contiguous slot algorithm can be effective.

Reference: MVS/ESA SP4.2 Working Set Management and Block Paging Presentation Guide, GG66-3204, pages 50-53.

MVS/ESA SP4.2 Program Code, LJB2-9605, routines IRARMPR5 and IRARMPR9 of module IRARMPRP.

Please note that while these references apply to MVS/ESA SP4.2, the finding is generally applicable after SP4.2.

"Paging Analysis in an Expanded Storage Environment", Beretvas, Thomas (IBM Corporation), *CMG '87 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL.

"Central Storage Management with MVS/ESA", Deese, Donald R. (Computer Management Sciences), *CMG '93 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL.

"Expanded Storage Management with MVS/ESA", Deese, Donald R. (Computer Management Sciences), *CMG '93 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM054: The number of page slots allocated to individual local page data sets may not be sufficient

Finding: CPExpert has determined that the number of page slots allocated to a particular local page data set may not always allow the Auxiliary Storage Manager to implement the *contiguous slot algorithm*. This finding applies only if a different number of page slots are allocated to **individual** local page data sets. This finding identifies specific local page data sets which have insufficient page slots assigned.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The level of impact depends upon how many physical swaps to auxiliary storage are done by the SRM and how much the average swap is delayed.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage significantly delayed the service class

Discussion: Swaps to auxiliary storage occur in "swap sets" of 30 page frames if local page data sets are used. The number of pages to be swapped out is divided into these swap sets. For example, if 95 page frames are swapped, there would be 4 swap sets. Three of the swap sets would be 30 page frames and 1 swap set would be 5 frames.

The Auxiliary Storage Manager (ASM) will attempt to attempt to write the swap set to a contiguous set of slots. The logic necessary to locate contiguous slots is called "the contiguous slot algorithm".

If the contiguous slot algorithm is successful in locating contiguous slots, only one seek is required to write the swap set (and only one seek would be required to read the swap set when the address space is swapped back in). Additionally, there would be no latency between page writes (or page reads upon swap-in).

The effect of the contiguous slot algorithm is to minimize the device service time per page and to minimize the device utilization.

The contiguous slot algorithm is effective only if there is a sufficiently large number of slots allocated to local page data sets so that fragmentation of the data sets does not preclude the ASM finding contiguous slots. As a

general guidance, the number of slots allocated should be at least four times the number of slots used.

CPExpert analyzes the number of slots allocated to local page data sets, versus the number of slots in use. This analysis is based upon information contained in SMF Type 75 records (SMF75SLA and SMF75MXU, respectively).

Rule **WLM052** will be produced if the **total** number of slots allocated for local page data sets is not four times the number of slots used and if some service class missed its performance goal because of page-in or swap-in delays.

However, you may have allocated a different number of page slots to different local page data sets. While the total number of slots may not cause the contiguous slot algorithm to be ineffective, the slots allocated to any individual page data set may be sufficiently low that the contiguous slot algorithm may not be effective for the particular local page data set.

Rule **WLM054** will be produced if a varying number of slots are allocated to local page data sets, and if the number of slots allocated for an **individual** local page data set is not four times the number of slots used by the local page data set.

The following example illustrates the output from Rule WLM054:

```
RULE WLM054:  THE NUMBER OF ALLOCATED PAGE SLOTS MAY BE INSUFFICIENT

   The number of slots allocated for the local page data set on VOLSER
   PG3040 may not always allow the Auxiliary Storage Manager to implement
   the contiguous slot algorithm.  During the measurement intervals shown
   below, the number of available slots was less than the desired percent
   of slots allocated, and at least one service class missed its performance
   goal because of paging delays from auxiliary storage.  As a general rule,
   the number of allocated slots should be about 4 times the average number
   of slots used to ensure that the contiguous slot algorithm has sufficient
   space.

                              AUXILIARY STORAGE      SLOTS       PERCENT
   MEASUREMENT INTERVAL        SLOTS ALLOCATED        USED      SLOTS USED
   7:30- 8:00, 16AUG1995            36,000          16,427        45.6
   8:00- 8:30, 16AUG1995            36,000          16,090        44.7
```

Suggestion: CPExpert suggests that you consider increasing the number of slots allocated to the local page data set so that the contiguous slot algorithm can be effective.

Additionally, you may wish to review the decision to allocate a varying number of slots to different local page data sets.

Reference: "Paging Analysis in an Expanded Storage Environment", Beretvas, Thomas (IBM Corporation), *CMG '87 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL. **This paper is highly recommended**.

Rule WLM055: Local page data sets are on same volume as swap data sets

Finding: CPExpert has determined that local page data sets are defined on the same volume as swap data sets.

Impact: This finding can have a LOW impact, MEDIUM impact, or HIGH impact on performance of your computer system. The level of impact depends upon the amount of page delay being experienced. This rule does not apply with OS/390 V2R10, as swap data sets are not supported beginning with this release of MVS.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage significantly delayed the service class

Discussion: Local page data sets generally should be allocated on separate volumes, and generally should not be allocated on volumes containing swap data sets nor should they be allocated on volumes containing other data sets which have a high level of activity. This is because (1) the device must perform seeks between the data sets, (2) the data transfer time to the device may significantly elongate resolution of page faults, and (3) the Auxiliary Storage Manager (ASM) will be unable to implement the suspend/resume function if you place more than one data set on the same volume.

- The device must perform seeks to position the arm between local page data sets and other data sets on the device. This seeking can be a major contributor to elongating the page fault resolution, since seeking typically requires 10-20 milliseconds per arm movement.

- Data transfer time to the device may significantly elongate page fault resolution. Swap data sets cause relatively large amounts of data to be transferred. The swaps are split into page groups of 12 pages if swap data sets are used. Each page group requires about 20 milliseconds of device activity to transfer the data (average latency plus data transfer time to transfer the page group).

- Each I/O request for another data set will interrupt the suspended I/O for the local page data set. The suspended I/O end, and must be restarted through the I/O Supervisor STARTIO function. Consequently, all potential performance gains resulting from the suspend/resume function of the ASM will be lost.

©Copyright 1994, Computer Management Sciences, Inc.

Please note that the suspend/resume function is not particularly important unless a relatively large amount of paging is done.

CPExpert produces Rule WLM055 only if a service class missed its performance goal and (1) page-in delay from auxiliary storage a major performance problem or (2) swap-in delay from auxiliary storage was a major performance problem.

The following example illustrates the output from Rule WLM055:

```
RULE WLM055:  LOCAL PAGE DATA SETS ARE ON SAME VOLUME AS SWAP DATA SETS

  CPExpert has determined that a local page data set and a swap data set
  are defined on VOLSER PG3041.  In most environments, allocating local
  page data sets and swap data sets on the same volume will result in
  overall poor performance of the paging subsystem.  In this case, page
  fault resolution from VOLSER PG3041 was significantly longer than the
  page fault resolution of other local page data sets.
```

Suggestion: CPExpert suggests that you separate the local page data sets and swap data sets onto different volumes, preferably on different paths.

Additionally, you may wish to reconsider the wisdom of defining swap data sets (see the discussion in Rule WLM061).

Reference: MVS Initialization and Tuning Guide
MVS/ESA SP5.1: Section 2.4 (Performance recommendations)

MVS Initialization and Tuning Guide
OS/390 MVS (V2R10): Changes

Please note that while this reference applies to MVS/ESA SP5 (Compatibility Mode), the finding is applicable to MVS/ESA SP5 (Goal Mode) and OS/390 (Goal Mode), except for OS/390 V2R10 and later versions of MVS.

Rule WLM056: Local page data sets share volume with Common or PLPA

Finding: CPExpert has determined that local page data sets are defined on the same volume as the Common page data set or the Pageable Link Pack Area (PLPA) page data set.

Impact: This finding can have a LOW impact, MEDIUM impact, or HIGH impact on performance of your computer system. The level of impact depends upon the amount of page delay being experienced.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: The delay time experienced by individual page fault resolution may increase significantly if local page data set are defined on the same volume as contains Common page data set or PLPA page data set. This is because the device must perform seeks between the page data sets, and seek time is significant.

Delays to page faults from Common or PLPA can delay many users. Thus, these delays are more serious than if the page fault were to the private area.

Additionally, if you place more than one page data set on the same volume, the Auxiliary Storage Manager (ASM) will be unable to implement the suspend/resume function. Each I/O request for one page data set will interrupt the suspended I/O for the other data set. The suspended I/O end, and must be restarted through the I/O Supervisor STARTIO function. Consequently, all potential performance gains resulting from the suspend/resume function will be lost.

Under some circumstances, performance would not be degraded by having Common and PLPA page data sets share a volume with a local page data set. For example, in a system with sufficient expanded storage (and perhaps with storage isolation protecting the Common area), there may be few or no page faults to Common or PLPA. CPExpert checks SMF Type 71 variables describing the pages read from and written to Common and PLPA (SMF71SNI and SMF71SNO, respectively) and does not produce Rule WLM056 if these counts are low.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert produces Rule WLM056 if some service class missed its performance goal because of page-in delays or swap-in delays, and a local page data set shared a volume with Common or PLPA.

The following example illustrates the output from Rule WLM056:

```
RULE WLM056:  LOCAL PAGE DATA SETS SHARE VOLUME WITH COMMON OR PLPA

   CPExpert has determined that a local page data set and either COMMON or
   PLPA page data sets are allocated on VOLSER PG3040.  In many environments,
   allocating local page data sets on the same volume as COMMON or PLPA
   will result in overall poor performance of the paging subsystem.  In
   this case, page fault resolution from VOLSER PG3040 was significantly
   longer than the page fault resolution of other local page data sets.
```

Suggestion: CPExpert suggests that you separate the Common/PLPA and local page data sets onto different volumes, preferably on different paths. Common and PLPA may share a volume only if the activity is low for these data sets (see the discussion in Rule WLM060).

Reference: MVS Initialization and Tuning Guide, MVS/ESA SP5.
Section 2.4 (Performance recommendations)

Please note that while this reference applies to MVS/ESA SP5 (Compatibility Mode), the finding is applicable to MVS/ESA SP5 (Goal Mode), OS/390 (Goal Mode), and z/OS (Goal Mode).

Rule WLM057: Multiple local page data sets are on same volume

Finding: CPExpert has determined that multiple local page data sets are defined on the same volume.

Impact: This finding can have a LOW impact, MEDIUM impact, or HIGH impact on performance of your computer system. The level of impact depends upon the amount of page delay being experienced.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage significantly delayed the service class

Discussion: The delay time experienced by individual page fault resolution will increase significantly if there is more than one local page data set on the same volume. This is because the device must perform seeks between the page data sets, and seek time is significant.

Additionally, if you place more than one page data set on the same volume, the Auxiliary Storage Manager (ASM) will be unable to implement the suspend/resume function. Each I/O request for one page data set will interrupt the suspended I/O for the other data set. The suspended I/O end, and must be restarted through the I/O Supervisor STARTIO function. Consequently, all potential performance gains resulting from the suspend/resume function will be lost.

CPExpert produces Rule WLM057 only if a service class missed its performance goal and (1) page-in delay from auxiliary storage a major performance problem or (2) swap-in delay from auxiliary storage was a major performance problem.

The following example illustrates the output from Rule WLM057:

```
RULE WLM057:  MULTIPLE LOCAL PAGE DATA SETS ARE ON THE SAME VOLUME

  CPExpert has determined that multiple local page data sets are defined
  on VOLSER PG3041.  In most environments, allocating multiple page data
  sets on the same volume will result in overall poor performance of the
  paging subsystem.
```

©Copyright 1994, Computer Management Sciences, Inc.

Suggestion: CPExpert suggests that you separate the local page data sets onto different volumes, preferably on different paths.

It is **not** necessary that the local page data sets be on dedicated paths, nor is it necessary that the paths to which they are assigned be lightly used. It is far better to split the local page data sets onto separate devices and paths and have their I/O operations compete with other system I/O, than it is to have the local page data sets reside on the same device!

Reference: MVS Initialization and Tuning Guide, MVS/ESA SP5.1
Section 2.4 (Performance recommendations)

Please note that while this reference applies to MVS/ESA SP5 (Compatibility Mode), the finding is applicable to MVS/ESA SP5 (Goal Mode), OS/390 (Goal Mode), and z/OS (Goal Mode).

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM058: Device response for local page data sets was significantly imbalanced

Finding: CPExpert has detected that the device response time for a local page data set was significantly worse than the average device response time for other local page data sets in the paging subsystem.

Impact: This finding can have a LOW impact, MEDIUM impact, or HIGH impact on performance of your computer system. The level of impact depends upon the amount of page delay being experienced.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: The device response time experienced by local page data sets will usually vary somewhat between DASD devices on which the local page data sets reside. This variance is typically caused by such factors as the number of pages transferred in a single I/O operation, the activity of other devices on the device controller, the activity of other devices on the path, etc.

A relatively minor variance in device response times between local page data sets is unavoidable.

However, if the device response time for an particular local page data set is significantly worse than the average for other local page data sets, there may be opportunities for performance improvement by eliminating the cause of the poor device response. These opportunities exist because the paging I/O is experiencing unusual contention at the device, controller, or path.

CPExpert analyzes the average device response time for all devices containing local page data sets. For each device, CPExpert computes the average device response time of **other** devices containing local page data sets.

CPExpert produces Rule WLM058 if the device response time for any device containing a local page data set was significantly worse than the average for the other local page data sets, and if page-in from auxiliary storage was a major performance problem for any service class.

The following example illustrates the output from Rule WLM058:

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM058:  LOCAL PAGE RESPONSE WAS SIGNIFICANTLY WORSE THAN AVERAGE

   CPExpert has detected that the device response time for a local page
   data set was significantly worse than the average device response time
   for other local page data sets.  This situation usually is caused by
   overloading the path to the local page data set or by placing local page
   data sets on volumes with other data sets.  During the below intervals,
   there was a significant imbalance of page transfer times among the local
   page data sets, and at least one service class missed its performance
   goal because of delays for page-in from auxiliary storage.

                                  AVERAGE     VOLUME              VOLUME
                                 PAGE XFR    WITH POOR   PAGES   AVG PAGE
   MEASUREMENT INTERVAL            TIME      XFR  TIME   XFR'D   XFR TIME
    8:00- 8:30, 16AUG1995         0.006       PG3040    10,875     0.010
```

Suggestion: CPExpert suggests that you determine and correct the cause of the poor device response time provided to the local page data set. These actions should be taken if the problem is persistent, or if the problem occurs at times when page response time is critical. (That is, you probably shouldn't worry if the problem doesn't happen often or if it happens only at relatively unimportant times.)

If you have licensed the DASD Component of CPExpert, you may wish to select the specific VOLSER for detailed analysis (using the SELECT option). If you do not license the DASD Component of CPExpert, you may be able to identify the cause of the problem by simple inference using standard RMF reports.

- You can examine the activity of other devices on the controller, using the RMF *Device Activity Report*. If the activity of these devices is high, then the problem probably is at the controller level. You may wish to move the local page data set to a different controller.

- You can examine the activity of all paths through which the device is accessed, using the RMF *Channel Path Activity Report*. If the activity of all paths is high (and the problem was not at the controller level), then the problem probably is at the path level. You may wish to move the local page data set to different paths (or provide more paths).

Please refer to the suggestions in Rule WLM400 for general ways to improve the performance of your paging subsystem.

You can alter the value CPExpert uses to assess whether page-in delay is unacceptable (unbalanced) by changed the OKPAGEIN value in USOURCE(WLMGUIDE).

Rule WLM059: Insufficient local page data sets are defined for migration

Finding: CPExpert has determined that the number of local page data sets is insufficient to allow parallel I/O operations for migration of pages from expanded storage to auxiliary storage.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The level of impact depends upon how often migration occurs.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: Please see the discussion contained in the referenced "Expanded Storage Management with MVS/ESA" regarding expanded storage operation. The discussion in this document is important for background information on the Migrator and other expanded storage concepts. Some of the relevant discussion is repeated below.

When the Migrator runs, it has a "migration quota" which it is trying to fill. This is simply the number of expanded storage frames which must be made available to alleviate the shortage. The quota of frames is the RCEAECOK threshold minus the RCEAECLO threshold. This quota differs depending upon whether the MVS version is SP4.2 or prior versions, and differs depending upon whether the Migrator was scheduled by the RSM or by the SRM.

- For pre-SP4.2, the initial low value (RCEAECLO) is 50 frames and the initial OK value (RCEAECOK) is 100 frames. For pre-SP4.2, the Migrator has a quota of at least 50 frames (100 - 50) which it will attempt to migrate.

- With SP4.2, the initial values are based on the MCCAECTH keyword in IEAOPTxx. The MCCAECTH has a default specification of (150,300). Thus, the default low value (RCEAECLO) is 150 frames and the OK value (RCEAECOK) is 300 frames. With SP4.2, the Migrator has a quota of at least 150 frames (300 - 150) which it will attempt to migrate.

- The initial RCEAECLO and RCEAECOK values can be adjusted based upon how often expanded storage becomes constrained. The RCEAECLO and RCEAECOK values can be increased when expanded

storage is constrained (so the Migrator will run earlier). These values subsequently can be lowered if expanded storage is no longer constrained (so the Migrator does not run as soon). The RCEAECLO and RCEAECOK values will not be lowered below their initial values. The migration quota will never be less than the amount computed based on the default values.

- The quota which the Migrator is attempting to fill can be larger than the default. This situation occurs if the SRM has scheduled the migration to accommodate the processor storage requirements of an address space which the SRM is attempting to swap in. The SRM can raise the RCEAECLO and RCEAECOK thresholds if necessary to make available more processor storage to contain the working set of a swapped-in address space.

It is important to appreciate that the Migrator does not migrate pages at a steady rate as might be reported by RMF, but migrates in "bursts" of pages. The "burst" will be the quota which the Migrator is attempting to fill. As the previous paragraphs discussed, the Migrator will have a quota of either 50 or 150 pages (or more), depending upon the version of MVS. This number of pages will be migrated during one "burst" of migration, even though RMF might report an average migration rate of only 10 pages per second (for example).

When the Migrator sends page-out requests to the Auxiliary Storage Manager (ASM), the Migrator sends the pages in page groups. The ASM will send these page groups out to local page data sets as parallel I/O operations, so long as there are sufficient local page data sets defined. As described above, the quota which the Migrator is attempting to reach is a minimum of 50 pages (pre-SP4.2) and 150 pages (with SP4.2), and this quota may be increased depending upon normal SRM adjustments.

A page group is 30 pages. Consequently, in order to take advantage of the significant performance benefits of parallel I/O operations for migration, a minimum of 2 local page data sets (50/30=2) should be defined for pre-SP4.2 environments. A minimum of 5 local page data sets (150/30=5) should be defined with SP4.2 and subsequent versions of MVS.

These are the **minimum** numbers based on the initial (default) values for the thresholds. However, the algorithms used by the SRM to adjust the RCEAECLO and RCEAECOK thresholds can increase the difference between the RCEAECOK and RCEAECLO values, depending upon how often expanded storage is constrained. Consequently, CPExpert believes that at least 3 local page data sets should be defined for pre-SP4.2 environments and at least 6 local page data sets should be defined for SP4.2 and subsequent versions of MVS.

©Copyright 1993, Computer Management Sciences, Inc.

From a performance viewpoint, it is important to remember that the migrated pages will be sent in page groups, and these page groups place a significant load on the local page data sets. Significant page-in delays could occur if page-in operations should be queued behind migration page groups.

The page-in delays might not be reflected in "average" data (for example, the delays might not be revealed by average TSO trivial response times over some RMF measurement interval). However, individual page-in operations (and individual TSO responses) could be significantly delayed. These delays could become noticeable even in average data if sufficient migration occurs during any interval.

To minimize the periodic performance problems, sufficient local page data sets must be defined.

CPExpert has detected that (1) a service class was significantly delayed because of paging from auxiliary storage, (2) fewer than the desired minimum local page data sets have been defined, and (3) more than a nominal amount of migration occurred.

Rule WLM059 is produced if less than six local page data sets have been defined.
It is possible that six local page data sets is still too few local page data sets. However, SMF records contain no information which can be used to determine whether MVS has increased the migration quota above the default such that more than six local pages data sets would be optimal.

The following example illustrates the output from Rule WLM059:

```
RULE WLM059:  INSUFFICIENT LOCAL PAGE DATA SETS ARE DEFINED FOR MIGRATION

   The number of local page data sets is insufficient to allow parallel
   I/O operations for migration of pages from expanded storage to auxiliary
   storage.  If migration occurs often, defining too few local page data
   sets could have a significant impact on paging performance.  During the
   measurement intervals shown below, migration was relatively high and at
   least one service class missed its performance goal because of paging
   delays from auxiliary storage.  CPExpert suggests increasing the number
   of local page data sets.

                                  AVERAGE     LOCAL PAGE DATA     SUGGESTED
   MEASUREMENT INTERVAL      MIGRATION RATE   SETS ALLOCATED    ALLOCATION
    7:30- 8:00, 16AUG1995         1.8                3              6
    8:00- 8:30, 16AUG1995         1.7                3              6
```

Suggestion: CPExpert suggests that you consider increasing the number of local page data sets so that migrated pages can be sent using parallel I/O operations. These local page data sets must be on separate paths, or the parallel I/O operations cannot be implemented.

It is not necessary that the local page data sets be the only data set on their volumes. Unless you have severe migration problems, it is unlikely that there will be a serious conflict between (1) the migration or subsequent page-in operations and (2) relatively low-activity data sets residing on the same volume as the local page data sets. It normally is far more important to allow the parallel I/O operations to be effective than it is to restrict the number of local page data sets simply because they must share the volumes with other data sets.

Reference: MVS/ESA SP4.2 Working Set Management and Block Paging Presentation Guide, GG66-3204, pages 50-54.

MVS (Real Storage Manager) source code, module IAREA.

"Expanded Storage Management with MVS/ESA", Deese, Donald R. (Computer Management Sciences), *CMG '93 Conference Proceedings*, The Computer Measurement Group, Inc., Chicago, IL.

Rule WLM060: PLPA and COMMON page data sets may be combined onto same actuator

Finding: CPExpert believes that the PLPA and COMMON page data sets may be combined onto the same actuator.

Impact: This finding can have a LOW IMPACT on performance of your computer system. However, the finding might save the cost of one actuator!

Logic flow: This is a basic finding. There are no predecessor rules. However, this finding is suppressed if any service class waited for Common page faults to be resolved from auxiliary storage.

Discussion: Paging for Pageable Pink Pack Area (PLPA) and Common Service Area (CSA) often is quite low. If paging is low for the PLPA and CSA, there is little reason to dedicate individual local page packs to the PLPA and COMMON page data sets. These data sets can be placed onto a single actuator with virtually no performance effect.

If the PLPA and COMMON page data sets are placed onto a single actuator, this could free one actuator for other use.

CPExpert produces Rule WLM060 if the number of page-in operations from PLPA and COMMON was extremely low and if no service classes waited for pages from COMMON.

The following example illustrates the output from Rule WLM060:

```
RULE WLM060:  PLPA AND COMMON PAGE DATA SETS MAY BE COMBINED

   CPExpert noticed that the page-in rate for the Pageable Link Pack Area
   (PLPA) and the Common area (COMMON) was so low during the entire period
   being analyzed that performance would not be adversely effected if the
   two data sets had been combined onto a single actuator.  Combining these
   low-activity data sets onto one actuator would make the second actuator
   available for other data sets.  Please make sure that this finding
   regularly occurs (perhaps after analyzing a week's performance data)
   before combining the data sets.
```

Suggestion: CPExpert suggests that you consider combining the PLPA and COMMON page data sets onto a single actuator. This is effectively accomplished by defining a **very** small PLPA page data set and increasing the size of the

COMMON page data set to accommodate the PLPA pages. PLPA will overflow into the COMMON page data set.

Please make sure that this finding regularly occurs (perhaps after analyzing a week's performance data) before combining the data sets.

Reference: "Analysis of Processor Storage and Paging Configurations," Bretvas, Thomas (IBM Corporation), *Conference Proceedings*, 1986 International Conference on Management and Performance Evaluation of Computer Systems (CMG '86).

Rule WLM061: Swap data sets are defined

Finding: CPExpert has determined that swap data sets are defined in the swap/page configuration.

Impact: This finding can have a MEDIUM IMPACT on performance of your computer system. This rule applies only to an environment with expanded storage. This rule does not apply with OS/390 V2R10, as swap data sets are not supported beginning with this release of MVS.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM400: Page-in from auxiliary storage was a major performance problem

Discussion: Swap data sets can be used by the Auxiliary Storage Manager (ASM) to contain Local System Queue Area (LSQA) and private area pages that are swapped in with the address space.

For systems with expanded storage, the RSM and SRM may divide the working set pages into a primary and secondary working set[1].

- **Primary working set**. The **primary working set** consists of LSQA pages, fixed pages, and one page from each virtual storage segment that is included in the working set[2].

  The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage.

  - The primary working set may be migrated to swap data sets if swap data sets are defined and if sufficient space exists on the swap data sets.

  - If swap data sets are not defined or if insufficient space exists on the swap data sets, the primary working set is migrated to local page data sets.

[1]This division is done only if the swap is to be done to expanded storage. If the swap is to be directly to auxiliary storage, the division is not done (a swap directly to auxiliary storage is called a **single stage swap**).

[2]The working set is composed of those address spaces with UIC of zero or one (and potentially an "enriched" working set with UIC greater than one if storage is not a constraint). A virtual storage segment is one megabyte of virtual storage.

©Copyright 1994, Computer Management Sciences, Inc.

- **Secondary working set**. The **secondary working set** consists all working set pages not included in the primary working set. These are most non-LSQA, non-fixed, working set pages. Notice that the secondary working set does not include swap trim pages[3]

The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage. The secondary working set will always be migrated to local page data sets.

There are several advantages to using **only** local page data sets, rather than a mixture of swap data sets and local page data sets.

- The ASM load balancing algorithm selects the local page data set with the best performance to receive a page group. This algorithm automatically helps correct performance problems if local page packs are on heavily loaded paths or if local page packs are not dedicated. The ASM does not apply the load balancing algorithm to swap data sets.

- With expanded storage, most of the migration paging (that is, the migration of the secondary working set and migration of swap trim pages) is automatically sent to local page data sets. Thus, most of the pages associated with a swap (either directly in the case of the secondary working set, or indirectly in the case of swap trim pages) will be sent to local page data sets regardless of whether swap data sets are used. Consequently, swap data sets tend to be under-utilized in an expanded storage environment.

- Overall system performance normally would be much better if the volumes which were defined as swap data set volumes were redefined for local page data sets. The local page data sets would individually have a lower average page rate since there would be more volumes available (that is, the paging load would be spread over more volumes).

  For example, suppose you had defined four local page data sets and two swap data sets. Performance would normally be significantly improved if you redefined the swap data sets as local page data sets, for a total of six local page data sets.

The following example illustrates the output from Rule WLM061:

[3]Swap trim pages are those pages trimmed from an address space before it is swapped out. The swap trim pages are the pages in central storage at swap time, which are not included in the working set. The swap trim pages may be sent to expanded if they meet the expanded storage criteria or they will be sent to auxiliary storage.

```
RULE WLM061:  SWAP DATA SETS ARE DEFINED

    CPExpert has determined that swap data sets are used for swapping.  Most
    performance analysts agree that swap data sets generally should not be
    used for swapping.  Rather, local page data sets should be used in most
    environments.  During the period being analyze, at least one service
    class missed its performance objective, and a major cause of delay to
    the service class was waiting for pages from auxiliary storage or a
    swap-in delay.
```

Suggestion: CPExpert suggests that you consider changing your swap data sets to local paging data sets.

There may be circumstances in which you should retain the swap data sets. For example, you may have very large swap sets in an environment without adequate expanded storage. You may wish to retain swap data sets to prevent critical page-in operations from being slowed by the I/O required to service large swap sets.

The following issues should considered:

- Delay of critical page-in operations is unlikely to exist in an expanded storage environment. Since only the primary working set may be migrated to swap data sets (unless the swap is a single stage swap), little advantage is gained by having swap data sets. That is, the secondary working set will always migrate to local page data sets and the secondary working set is usually significantly larger than the primary working set. Since only the primary working set would be migrated, only the primary working set would be effected by having swap data sets.

- You normally should have sufficient local page data sets such that the ASM can initiate swap-out I/O operations in parallel to local page data sets. If the I/O operations are initiated in parallel, then the maximum delay to page-in operations normally would be only the time required to transfer a page group (30 pages for local page data sets).

  The time to transfer a page group normally would be about 50-60 milliseconds for an IBM-3380 paging device (the possible seek operation, search operation, and data transfer), and these times would become significantly less if the DASD were cached or if IBM-3390 devices were used for paging. This periodic delay would be offset if the swap data sets were converted to local page data sets, since more local page data sets would result in a lower average page-in time.

- Under some circumstances, the migration rate may be high. If the migration rate is high, one implication is that there are few available

pages in expanded storage. (The only purpose of migrating pages is because there is an insufficient number of available expanded storage pages.)

If there are few available expanded storage pages, the SRM will direct swaps to auxiliary as **single-stage** swaps, and will not prepare a primary and secondary working set. In this situation, allocating swap data sets may prevent the single-stage swaps from overloading the local page data sets.

Of course, if many swaps are sent to auxiliary rather than to expanded, you have basic problems with your expanded storage environment.

Samson (see references) gives additional examples in which you may wish to retain swap data sets.

Unless you have a very unusual situation, CPExpert suggests that you eliminate the swap data sets and (if appropriate) use the DASD volumes as additional local page data sets.

Reference: MVS Initialization and Tuning Guide
MVS/ESA: Section 2.4 (Performance recommendations)
MVS/ESA: Section 2.6 (question regarding expanded storage)

OS/390 MVS: Section 2.4 (Performance recommendations)
OS/390 MVS: Section 2.6 (question regarding expanded storage)

OS/390 MVS (V2R10): Changes

"Paging Analysis in an Expanded Storage Environment," Bretvas, Thomas (IBM Corporation), *Conference Proceedings*, 1987 International Conference on Management and Performance Evaluation of Computer Systems (CMG '87). **This paper is highly recommended!**

MVS Performance Management, Samsom, Stephen L., McGraw-Hill Publishing Company (pages 241-242)

MVS System Resources Manager module IRARMST2.

Rule WLM070: Terminal Output Wait swaps occur too often

Finding: CPExpert has determined that an excessive number of Terminal Output Wait swaps occurred during the measurement interval.

Impact: This finding can have a LOW IMPACT or MEDIUM IMPACT on performance of your computer system. The level of impact depends upon how many Terminal Output Wait swaps occurred and upon your management objectives.

Logic flow: The following rule causes this rule to be invoked:
WLM450: Swap-in delay helped cause service class to miss goal

Discussion: Terminal Output Wait swaps occur after the SRM has been notified that a TSO session is in terminal wait after issuing a TPUT, and the address space is in a long wait condition. Terminal Output Wait swaps occur for a variety of reasons: (1) waiting for terminal output buffers, (2) TSO logon processing, (3) TSO batch SUBMIT, (4) entering and leaving ISPF, and (5) others.

- Terminal Output Wait swaps can occur because transactions are waiting for terminal output buffers. This means that TSO commands or CLISTs have produced output so fast that the TIOC does not have sufficient buffers to handle the terminal output. This cause has been touted as the primary cause of Terminal Output Wait swaps in many articles and journals[1], and CPExpert earlier considered insufficient terminal buffers to be the cause of Terminal Output Waits swaps.

- A Terminal Output Wait swap occurs whenever a TSO user issues a TSO batch SUBMIT command. It is unclear why this occurs, but the fact that a Terminal Output Wait swap follows a TSO batch SUBMIT command has been verified using GTF trace of SRM SYSEVENTs.

---

[1]As examples:

- "OS/VS2/MVS Resource Measurement Facility (RMF) Reference and User's Guide", publication SC28-0922, IBM Corporation.
- "MVS/XA Performance and Resource Measurement Facility (RMF) Data Analysis", J. William Mullen, *CMG'88 Conference Proceedings*, The Computer Measurement Group, Chicago, IL.
- "Cheryl Watson's TUNING Letter", January 1991, Watson & Walker, Inc., Lutz, FL

©Copyright 1994, Computer Management Sciences, Inc.

- Terminal Output Wait swaps can occur during TSO logon processing, if the terminal control address space (TCAS) program has not been marked non-swappable in the Program Properties Table in SCHED00.

  When the TCAS is swappable, the SRM will often find that the address space has not been executed for a period of time, and cause the address space to be swapped out for a Detected Wait Swap.

  If the TCAS program is swapped out for a Detected Wait Swap when a TSO user attempts to logon, the TSO user goes into a long wait and the SRM will swap the TSO user out for a Terminal Output Wait swap. The TCAS will be marked Ready and be swapped in to service the logon. Then the TSO user will be swapped in to perform whatever action is next appropriate.

- Terminal Output Wait swaps may occur if the VTAM **IOBUF** buffer pool start options are specified incorrectly. The IOBUF values describe how VTAM will manage the message pool in fixed storage. The **baseno** value establishes the base number of buffers in the pool. If this value is too low, VTAM will expand the buffer pool as required, and subsequently contract the pool. Any TSO user causing VTAM to expand the message pool will be swapped out for a Terminal Output Wait swap.

- Terminal Output Wait swaps occur whenever a TSO user enters or leaves ISPF.

- Terminal Output Wait swaps may occur whenever a TSO users operating under a multi-session manager transfers to another session. (We have been unable to verify that Terminal Output Waits result from this situation. However, preliminary analysis of GTF Trace of SRM SYSEVENTs indicate that Terminal Output Wait swaps may occur in this situation.)

The first three situations listed above are correctable. There apparently is nothing that can be done about the other situations.

CPExpert produces Rule WLM070 if swap-in delay was a significant factor in causing a service class to miss its performance goal, if the service class related to TSO, if Terminal Output Wait swaps accounted for a significant percent of the total swaps, and if Terminal Output Wait swaps exceed the **SWAPTO** guidance variable in USOURCE(WLMGUIDE).

Suggestion: CPExpert suggests you consider the following alternatives:

- **Verify that TSOKEYxx parameters are appropriate**. Terminal Output Wait swaps occur because transactions are waiting for terminal output buffers. This means that TSO commands or CLISTs have produced output so fast that the TIOC does not have sufficient buffers to handle the terminal output. You can determine whether this is a likely cause of the Terminal Output Wait swaps by examining the TSOKEYxx member of SYS1.PARMLIB.

  TSO terminal output buffers are controlled by three keywords in the TSOKEYxx member of SYS1.PARMLIB: BUFRSIZE, HIBFREXT, and LOBFREXT.

  - The BUFRSIZE value specifies the size in bytes of a VTIOC buffer. Input and output data smaller than the BUFRSIZE value use storage equal to the BUFRSIZE value. Buffers are dynamically allocated equal to the data size if input or output data is greater than the BUFRSIZE value.

    If the BUFRSIZE value is too large compared with the actual size of terminal data, central storage will be wasted while the buffers are used. For example, a LISTC command generates 80-character output. Suppose that the BUFRSIZE value were specified as 2048 (this size has been recommended by some authors[2]). Each buffer would waste 1968 (2048 - 80) bytes while the buffer was holding data. (This amount of wasted storage is not generally significant, considering the vast amount of storage available in modern systems.)

    If the BUFRSIZE value is too small compared with the actual size of the terminal data, storage will be dynamically allocated (using a GETMAIN). This dynamical allocation conserves storage, but uses processor resources to execute the GETMAIN and subsequent FREEMAIN.

    The BUFRSIZE value generally should be large enough to accommodate the buffer size requirements of 80% to 90% of your TPUTs.

    - Unless virtual storage is a constraint, it is better to specify a relatively large value for the BUFRSIZE value. A large value will avoid unnecessary GETMAIN/FREEMAIN processing.

      With most modern systems, virtual storage is not a constraint for TSO environments. Consequently, CPExpert normally suggests

[2]For example, "Cheryl Watson's TUNING Letter", January 1991, Watson & Walker, Inc., Lutz, FL

that a relatively large value (e.g., 2048) be specified for the BUFRSIZE value.

- If virtual storage is a constraint, it is better to specify a relatively small value and avoid wasting virtual storage. This situation is extremely unlikely.

- The HIBFREXT value specifies the maximum amount of virtual storage that can be dynamically allocated for output data. When this value is reached, no more output requests are honored until the LOBFREXT value is reached. The default HIBFREXT value in MVS/XA and MVS/ESA is 48000 bytes.

  The HIBFREXT value is used in conjunction with the BUFRSIZE value to determine the maximum number of buffers that can be allocated to a user (HIBFREXT ÷ BUFRSIZE = maximum number of buffers).

  If the HIBFREXT value is too low, the buffers can become exhausted and generate unnecessary swaps. For example, consider a BUFRSIZE of 2048 as mentioned above. If the HIBFREXT value were at the default of 48000 bytes, only 23 (48000 ÷ 2048) TPUTs could be serviced before a user would be swapped out for a Terminal Output Wait swap.

  Suppose that a LISTC command generated 50 lines. Each line would require a buffer, and only 23 buffers would fill before the user would be swapped out for Terminal Output Wait. The user would not even fill the standard 24-line screen before being swapped out. If the BUFRSIZE value were at its default of 132, then 363 buffers (48000 ÷ 132) would be available (assuming that no TPUTs were greater than the 132).

  If the HIBFREXT value is too high, then virtual storage could be wasted. However, keep in mind that this is virtual storage and virtual storage is rarely a constraint to an individual TSO user.

  CPExpert suggests that you consider increasing the HIBFREXT value to accommodate a **significant** number of terminal output buffers. Many installations increase the HIBFREXT value to 96,000 bytes or higher. Recall that this specification is **virtual storage**, and virtual storage will not be a constraint for many TSO address spaces. As described above, significant performance degradation can result from specifying too low a value for HIBFREXT.

- LOBFREXT specifies the minimum number of virtual storage bytes that can be dynamically allocated for output data. When this value is reached, tasks that are swapped out for Terminal Output Wait swaps are marked dispatchable. The default value for LOBFREXT in MVS/XA and MVS/ESA is 24000 bytes.

  If the LOBFREXT value is too high, a user swapped out for Terminal Output Wait swap will be marked ready and swapped in after only a few buffers have been emptied.

  For example, suppose that the BUFRSIZE value were specified as 2048 (as mentioned above), and that the default HIBFREXT and LOBFREXT values were used (these are 48000 and 24000 bytes, respectively).

  Suppose that a LISTC command generated 50 lines. Each line would require a buffer. As described above, only 23 buffers would fill before the user would be swapped out for Terminal Output Wait. The user would be marked Ready and swapped back in after only 12 buffers were emptied (48000 - (12 * 2048) < 24000). These 12 buffers would fill and the user would again be swapped out. The user would experience a total of 3 Terminal Output Wait swaps in order to generate the 50 lines (initial 23 lines followed by swap out, 12 more lines followed by swap out, 12 more lines followed by swap out)!

  As another example, the Broadcast data set often will send many lines to a user during logon, and these are sent in line mode. Each line will require a buffer and the number of buffers could easily be exhausted. With inappropriate specification for HIBFREXT and LOBFREXT values, a TSO user could experience **several** Terminal Output Wait swaps during logon.

  The LOBFREXT value should be small to minimize the repeated swapping of users. If LOBFREXT is too high, swapped out address spaces are marked ready and are swapped in, only to fill a few output buffers (this was illustrated in the above discussion). Swap delay time (time from when a swapped out user is marked ready until the user is swapped in) generally is fairly short. This particularly is true for interactive TSO users who probably are logically swapped rather than physically swapped.

  The swap-in delay should be **extremely** short, as the Workload Manager will manage the MPL constraints of the domain serving the TSO service class and normally will ensure that TSO users are not delayed by swap-in. Few users should be intolerant of such short delays. In fact, the only reason to have a LOBFREXT greater than

zero is to reduce the delay from the time a user is finished with one screen of output until the user receives the next screen of output.

Consider adjusting the LOBFREXT value to accommodate only 1 or 2 buffers. If expanded storage is installed, swap-in delay should be significantly less (the Terminal Output Wait swaps will almost certainly be directed to expanded storage). Therefore, consider specifying a LOBFREXT value of zero with expanded storage.

In summary, CPExpert suggests that you consider specifying the following values for the TSOKEYxx keywords for MVS/ESA environments:

BUFRSIZE = 2048
HIBFREXT = 96000
LOBFREXT = 0

These values should, of course, be altered for special circumstances.

- The suggested BUFRSIZE value generally should be set to a size sufficient to accommodate the buffer size requirements of 80% to 90% of your TPUTs.

- The HIBFREXT value could be significantly increased (e.g., doubled) if Terminal Output Wait swaps are still a significant percent of terminal wait swaps.

- The LOBFREXT value should be increased to either the SCRSIZE value or the BUFRSIZE value if there are unacceptable delays to a terminal user.

Some systems programmers are reluctant to significantly increase the TSOKEYxx buffer values. A part of this reluctance may be caused by remembering earlier systems when real and virtual storage were significant constraints, and when the TSO user community was quite small.

Even if real storage should be a constraint, it generally is better to incur some modest increased paging if the savings is a reduction in swapping. Swaps are **much** more expensive than paging operations.

- **Verify that TCAS is non-swappable**. The default Program Properties Table (module IEFSDPPT in SYS1.LINKLIB) specifies that TCAS is "privileged" but that it is swappable. This means that TCAS will be swapped when the SRM detects that the address space is in a long wait.

This default is specified because TSO is not used at all sites or is not a high priority at some IBM sites.

If you have a significant number of TSO logons and each logon generates two swaps (the Detected Wait and the Terminal Output Wait), you can save the swaps by making TCAS non-swappable. TCAS will still be subject to normal page stealing, so the storage is not completely reserved to TCAS.

Modify the SCHEDxx member of SYS1.PARMLIB by including the following:

```
PPT  PGNNAME(IKTCAS00)
     KEY(6)
     NOCANCEL
     NOSWAP
     PRIV
     SYST
```

The above causes the IKTCAS00 terminal control address space program to be non-swappable.

Note that unless your system is extremely storage-constrained so that logically swapping is not being used often, the TCAS program is only logically swapped. It is using central storage, even though it is swapped out. You are simply incurring the overhead of the Detected Wait swaps and Terminal Output Wait swaps needlessly.

- **Verify VTAM IOBUF pool definitions**. You should check the VTAM **IOBUF** pool definitions to determine whether the values are sufficient to prevent unnecessary expansion of the pool.

Reference: MVS Initialization and Tuning Reference
MVS/ESA SP5.1: Chapter 3.60
MVS/ESA SP5.2: Chapter 3.60
OS/390 (V1R1): Chapter 3.61
OS/390 (V1R2): Chapter 3.61
OS/390 (V1R3): Chapter 3.61
OS/390 (V2R4): Chapter 4.17
OS/390 (V2R5): Chapter 66.4
OS/390 (V2R6): Chapter 67.4
OS/390 (V2R7): Chapter 68.4
OS/390 (V2R8): Chapter 68.4
OS/390 (V2R9): Chapter 68.4

OS/390 (V2R10): Chapter 67.4
z/OS (V1R1): Chapter 67.4
z/OS (V1R2): Chapter 67.4
z/OS (V1R3): Chapter 68.4
z/OS (V1R4): Chapter 68.4

VTAM Installation and Resource Definition, pages 294 to 300

MVS Performance Notebook, page 246

VTAM Installation and Resource Definition, pages 294 to 300

MVS Performance Notebook, page 246

Rule WLM071: Detected Wait swaps occur too often

Finding: CPExpert has determined that an excessive number of Detected Wait swaps occurred during the measurement interval.

Impact: This finding can have a LOW IMPACT or MEDIUM IMPACT on performance of your computer system. The level of impact depends upon how many Detected Wait swaps occurred and upon your management objectives.

Logic flow: The following rule causes this rule to be invoked:

WLM450: Swap-in delay helped cause service class to miss goal

Discussion: Detected Wait swaps occur because the SRM detects that a resident transaction has not been dispatchable for two seconds of real time or eight SRM seconds, without issuing the WAIT,LONG=YES macro. (Eight SRM seconds represent about 320 milliseconds on an IBM-3090/600S single image or 865 milliseconds on an IBM-3084Q. Consequently, the eight SRM seconds will cause a detected wait on most of today's systems.) Detected Wait swaps usually are caused by cross memory services, applications that treats the terminal as SYSIN or SYSPRINT, teleprocessing applications (e.g., test CICS regions) that are not marked non-swappable, etc.

Additionally, STIMER wait values of less than the 0.5 seconds required to trigger a Long Wait swap may trigger a Detected Wait swap if the wait time is more than 8 SRM seconds (this situation is unlikely, but could occur on IBM-3090/180 and above).

The basic purpose of Detected Wait swaps is to detect address spaces which are using central storage, but which are not dispatchable. If the address spaces are not dispatchable, perhaps their central storage should be made available for other purposes.

From this definition of Detected Wait swaps, these swaps generally should be a fairly small percentage of the overall swaps, unless TSO is used as a gateway to some other application. However, Detected Wait swaps commonly account for almost 5% of the total swaps, and sometimes account for over 30% of the total swaps. A large percent of Detected Wait swaps should be investigated, since these applications tie up real storage before the SRM detects that they have not been dispatchable.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert produces Rule WLM071 if swap-in delay was a significant factor in causing a service class to miss its performance goal, if Detected Wait swaps accounted for a significant percent of the total swaps, and if the number of Detected Wait swaps exceeded the **SWAPDW** guidance variable in USOURCE(WLMGUIDE).

Suggestion: CPExpert recommends that you examine the specific applications causing the Detected Wait swaps. These applications are not dispatchable for an extended period of time, but do not advise the SRM so that they can be swapped out when they become non-dispatchable. They occupy storage for the period between when they become non-dispatchable and the SRM detects this status. This central storage could be added to the Available Frame Queue and made available to executing address spaces.

Alternatively, you may wish to review whether the address spaces should be swappable. Under some conditions, it may be better to make the address spaces non-swappable and reduce the Detected Wait swaps.

You can determine the workload experiencing the Detected Wait swaps by examining the RMF Monitor II "Address Space State Data Report". This report shows the reason for the last swap out for the jobs.

- You can select class, status, and domain. You should use the command **ASD(A,A)** to request a report of all address spaces that are currently executing, non-swappable, or swapped out and eligible for swap-in.

- Review the reason for last swap out. This information will be displayed in the "R LS" column. The reason codes are shown in the RMF; search for reason "DW". These identify address spaces experiencing Detected Wait swaps.

Detected Wait swaps seem to be application dependent, with little guidance available in the literature about how to prevent Detected Wait swaps.

We would appreciate your feed-back regarding the causes and cures of Detected Wait swaps in different applications! We can make this information generally available to users of CPExpert.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM080: JES-managed and WLM-managed job classes conflict

Finding: CPExpert has detected that a service class included both JES-managed job classes and WLM-managed job classes.

Impact: This finding should be viewed a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the batch jobs assigned to the service class. The level of impact will depend primarily on the number of JES-managed and WLM-managed jobs executing in the conflicting job classes.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: With OS/390 Version 2 Release 4, the Workload Manager allows installations to define job classes as being managed by Job Entry Subsystem (JES) or by the Workload Manager (WLM).

- For jobs assigned to JES-managed job classes, the normal selection of the jobs for initiation will be done. That is, JES initiators will select jobs from the job class queue based on the normal selection criteria (e.g., priority, aging, resource affinity, etc.).

- For jobs assigned to WLM-managed job classes, the WLM will control selection of jobs by dynamically changing the number of WLM initiators and/or their work selection criteria. These actions will be taken by the WLM in an attempt to met installation defined goals for the service classes to which the jobs are classified.

WLM control of batch work is enabled by changing the mode of a JES2 job class (on a job class by job class basis) to MODE=WLM. When that is done, no job in that JES2 job class will be selected by normal JES2 initiators. Instead WLM will be informed of the jobs waiting execution and will start and stop WLM controlled initiators based the current backlog of work.

After conversion of a job's JCL, JES2 places the job into an appropriate class queue to await execution. If the job class is a JES2 managed class (JOBCLASS MODE=JES), JES2 initiators select a job from the class queue (depending on the job's priority) and pass control to MVS to execute your program. If the job class is a WLM managed class (JOBCLASS MODE=WLM), JES provides the WLM with a list of jobs waiting to execute, by service class. Based on installation goals for the service classes, the WLM can determine whether job queue time is a significant delay to a

©Copyright 1994, Computer Management Sciences, Inc.

service class meeting its performance goal. If so, the WLM might start an initiator for a job class assigned to the service class.

Job initiators for WLM managed job classes are controlled dynamically by workload management. These initiators run under the Master Subsystem and are not assigned JES2 job numbers. WLM can adjust the number of initiators on each system based on:

- The queue of jobs awaiting execution in WLM managed classes.
- The performance goals and relative importance of this work.
- The success of meeting these goals.
- The capacity of each system to do more work.

The WLM evaluates the performance of service class periods relative to the performance goals established by the installation. It is important to appreciate that the performance of a service class period *is based on the work executing in the service class*. Further, the delays to work in the service class *is based on all work executing in the service class*. If the service class consists of jobs from JES-managed queues and jobs from WLM-managed queues, the WLM will be unable to distinguish delays caused by JES-managed jobs from those caused by WLM-managed jobs. Further, the JES-managed jobs could (and probably <u>would</u>) have different execution characteristics from the WLM-managed jobs.

When the WLM examines service class delays, it attempts to manage resource allocation to eliminate the most serious delays, where eliminating the delay would cause performance to be significantly improved.

If the work executing in the service class consists of JES-managed jobs (for which the WLM has no initiator control) and WLM-managed jobs (for which the WLM does have initiator control), the WLM will be unable to effectively determine whether execution queue delay would significantly improve performance of the work executing in a service class.

The OS/390 MVS Planning: Workload Management document specifically states that: *"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."*

Prior to OS/390 Version 2 Release 9, only the SMF Type 26 records contained an indication as to whether a job was assigned a job class

with MODE=JES or MODE=WLM. Since SMF Type 26 records are not often kept in a performance data base (and are not available until after a job has ended) analysis of conflicts between JES-managed and WLM-managed initiators was not feasible. Consequently, CPExpert requested that IBM place an indicator in the SMF Type 30 records so that potential problems with WLM-managed initiators could be analyzed. With OS/390 Version 2 Release 9, IBM created the SMF30WMI indicator as a part of the SMF30PF1 (performance section flag byte) variable.

CPExpert examines the SMF30WMI indicator in SMF Type 30 records to determine whether a job is assigned to a JES-managed or WLM-managed initiator. CPExpert then determines the service classes to which the jobs are assigned.

CPExpert produces Rule WLM080 if any jobs with MODE=JES are assigned to the same service class as jobs with MODE=WLM.

The following example illustrates the output from Rule WLM080:

```
RULE WLM080: JES-MANAGED AND WLM-MANAGED JOB CLASSES CONFLICT

   CPExpert detected that a service class included both JES-managed job
   classes and WLM-managed job classes.  The Workload Manager might not
   effectively meet its goals of managing initiators if some jobs are
   initiated through JES and other jobs are initiated through the Workload
   Manager.  The following information describes the jobs that executed
   and whether they were assigned to a job class initiated by JES or to
   a job class initiated by the Workload Manager.  Additionally, the
   information shows whether an operator changed the service class to
   which the job was assigned.

              JOB  SERVICE
   JOB NAME CLASS  CLASS  INIT INITIATOR TIME     SYSTEM  OPERATOR ACTION
   CQQUERY    R   DISCR    WLM 17MAR2000:15:01:09  J90
   CQQUERY    R   DISCR    WLM 17MAR2000:15:15:29  J90
   CQCHKPT    R   DISCR    WLM 17MAR2000:15:15:34  J90
   AS151810   K   DISCR    JES 17MAR2000:15:18:19  JG0    SRV CLS CHG BEFORE INIT
```

The OPERATOR ACTION information provided with Rule WLM080 will include:

- Job service class association was modified by a system operator prior to job initiation.
- Job service class association was modified by a system operator during job initiation.
- Job service initiation was forced by a system operator.
- Job has been restarted.

Suggestion: CPExpert suggests that you change the workload classification scheme or change the job class MODE definition to eliminate the conflicts between jobs assigned to JES-managed and WLM-managed initiators.

Reference: MVS Planning: Workload Management

OS/390 (V2R4): Section 3.3: Batch Workload Management
OS/390 (V2R5): Section 3.3: Batch Workload Management
OS/390 (V2R6): Section 3.3: Batch Workload Management
OS/390 (V2R7): Section 3.3: Batch Workload Management
OS/390 (V2R8): Section 3.3: Batch Workload Management
OS/390 (V2R9): Section 3.3: Batch Workload Management
OS/390 (V2R10): Section 3.3: Batch Workload Management
z/OS (V1R1): Section 3.3: Batch Workload Management
z/OS (V1R2): Section 3.3: Batch Workload Management
z/OS (V1R3): Section 3.3: Batch Workload Management
z/OS (V1R4): Section 3.3: Batch Workload Management

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM081: WLM-managed job class assigned to multiple service classes

Finding: CPExpert has detected that a WLM-managed job class was assigned to more than one service class based on the workload classification scheme.

Impact: This finding should be viewed a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the batch jobs assigned to the service class. The level of impact will depend primarily on the number of WLM-managed jobs executing in the multiple service classes.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: With OS/390 Version 2 Release 4, the Workload Manager allows installations to define job classes as being managed by Job Entry Subsystem (JES) or by the Workload Manager (WLM).

- For jobs assigned to JES-managed job classes, the normal selection of the jobs for initiation will be done. That is, JES initiators will select jobs from the job class queue based on the normal selection criteria (e.g., priority, aging, resource affinity, etc.).

- For jobs assigned to WLM-managed job classes, the WLM will control selection of jobs by dynamically changing the number of WLM initiators and/or their work selection criteria. These actions will be taken by the WLM in an attempt to met installation defined goals for the service classes to which the jobs are classified.

WLM control of batch work is enabled by changing the mode of a JES2 job class (on a job class by job class basis) to MODE=WLM. When that is done, no job in that JES2 job class will be selected by normal JES2 initiators. Instead WLM will be informed of the jobs waiting execution and will start and stop WLM controlled initiators based the current backlog of work.

After conversion of a job's JCL, JES2 places the job into an appropriate class queue to await execution. If the job class is a JES2 managed class (JOBCLASS MODE=JES), JES2 initiators select a job from the class queue (depending on the job's priority) and pass control to MVS to execute your program. If the job class is a WLM managed class (JOBCLASS MODE=WLM), JES provides the WLM with a list of jobs waiting to execute, by service class. Based on installation goals for the service classes, the WLM can determine whether job queue time is a significant delay to a

©Copyright 1994, Computer Management Sciences, Inc.

service class meeting its performance goal. If so, the WLM might start an initiator for a job class assigned to the service class.

Job initiators for WLM managed job classes are controlled dynamically by workload management. These initiators run under the Master Subsystem and are not assigned JES2 job numbers. WLM can adjust the number of initiators on each system based on:

- The queue of jobs awaiting execution in WLM managed classes.
- The performance goals and relative importance of this work.
- The success of meeting these goals.
- The capacity of each system to do more work.

The WLM evaluates the performance of service class periods relative to the performance goals established by the installation. It is important to appreciate that the performance of a service class period *is based on the work executing in the service class*. Further, the delays to work in the service class *is based on all work executing in the service class*.

When the WLM examines service class delays, it attempts to manage resource allocation to eliminate the most serious delays, where eliminating the delay would cause performance to be significantly improved.

If the WLM workload classification scheme results in jobs assigned to a job class with MODE=WLM being assigned to more than one service class, the WLM will be unable to distinguish the effect of execution queue delay on the performance of the multiple service classes.

The execution queue delay for the jobs will be included in the calculation of execution velocity or response time for the service classes. The resulting execution velocity or response time will be used to calculate a performance index for the service classes. The resulting performance index will be used by the WLM to assess how well the service classes are meeting their performance goals. If a service class is not meeting its performance goal, the WLM might decide to add an initiator to the job class to attempt to reduce the execution queue delay.

However, the initiator will be associated with the job class queue. Since jobs in the job class queue are assigned to more than one service class, the WLM would be unable to take action that would specifically address execution queue delays for the service class missing its goal. Consequently, the WLM would not be effective in managing the initiators for the job class.

©Copyright 1994, Computer Management Sciences, Inc.

Prior to OS/390 Version 2 Release 9, only the SMF Type 26 records contained an indication as to whether a job was assigned a job class with MODE=JES or MODE=WLM. Since SMF Type 26 records are not often kept in a performance data base (and are not available until after a job has ended) analysis of conflicts between JES-managed and WLM-managed initiators was not feasible. Consequently, CPExpert requested that IBM place an indicator in the SMF Type 30 records so that potential problems with WLM-managed initiators could be analyzed. With OS/390 Version 2 Release 9, IBM created the SMF30WMI indicator as a part of the SMF30PF1 (performance section flag byte) variable.

CPExpert examines the SMF30WMI indicator in SMF Type 30 records to determine whether a job is assigned to a JES-managed or WLM-managed initiator. CPExpert then determines the service classes to which the jobs in a job class with MODE=WLM are assigned.

CPExpert produces Rule WLM081 if any jobs assigned to a job class with MODE=WLM are assigned to the more than one service class.

The following example illustrates the output from Rule WLM081:

```
RULE WLM081: WLM-MANAGED JOB CLASS ASSIGNED TO MULTIPLE SERVICE CLASSES

   CPExpert detected that a job class with MODE=WLM was classified to
   more than one service class.  The Workload Manager bases many of its
   decisions about WLM-managed initiators on the achievement of goals for
   the service classes to which the jobs are assigned. The Workload
   Manager might not effectively manage initiators if jobs are assigned
   to service classes with different goals or goal importance.
   Additionally, the Workload Manager will be unable to honor job class
   limits, because the WLM considers job class limits only with a service
   class.  Consequently, the WLM might start extra initiators even though
   the job class limits would restrict jobs from running.

                      JOB
   JOB NAME          CLASS    SYSTEM   SERVICE CLASS   INITIATOR TIME
   CQQUERY             R      J90      DISCR           17MAR2000:15:01:09
   CQQUERY             R      J90      DISCR           17MAR2000:15:15:29
   CQCHKPT             R      J90      DISCR           17MAR2000:15:15:34
   TEST1               R      J90      TEST081         17MAR2000:15:01:09
   TEST2               R      J90      TEST081         17MAR2000:15:15:29
   TEST3               R      J90      TEST085         17MAR2000:15:01:09
```

Suggestion: CPExpert suggests that you (1) change the workload classification scheme to eliminate the assignment of jobs in the job class to more than one service class or (2) change the job classification scheme so the jobs are assigned to different jobs classes. Either option should result in the specific jobs in the job class shown in the output from Rule WLM081 being assigned to the same service class or assigned to different job classes.

Reference: MVS Planning: Workload Management

OS/390 (V2R4):	Section 3.3: Batch Workload Management
OS/390 (V2R5):	Section 3.3: Batch Workload Management
OS/390 (V2R6):	Section 3.3: Batch Workload Management
OS/390 (V2R7):	Section 3.3: Batch Workload Management
OS/390 (V2R8):	Section 3.3: Batch Workload Management
OS/390 (V2R9):	Section 3.3: Batch Workload Management
OS/390 (V2R10:	Section 3.3: Batch Workload Management
z/OS (V1R1):	Section 3.3: Batch Workload Management
z/OS (V1R2):	Section 3.3: Batch Workload Management
z/OS (V1R3):	Section 3.3: Batch Workload Management
z/OS (V1R4):	Section 3.3: Batch Workload Management

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM082: Job might not be suitable for WLM-managed initiators

Finding: CPExpert has detected that a WLM-managed job was spent a major portion of its total active time in a queue waiting for an initiator.

Impact: This finding should be viewed a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the batch jobs assigned to the service class. The level of impact will depend primarily on the number of WLM-managed jobs that spent a major portion of their active time in a queue waiting for an initiator.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: With OS/390 Version 2 Release 4, the Workload Manager allows installations to define job classes as being managed by Job Entry Subsystem (JES) or by the Workload Manager (WLM).

- For jobs assigned to JES-managed job classes, the normal selection of the jobs for initiation will be done. That is, JES initiators will select jobs from the job class queue based on the normal selection criteria (e.g., priority, aging, resource affinity, etc.).

- For jobs assigned to WLM-managed job classes, the WLM will control selection of jobs by dynamically changing the number of WLM initiators and/or their work selection criteria. These actions will be taken by the WLM in an attempt to met installation defined goals for the service classes to which the jobs are classified.

WLM control of batch work is enabled by changing the mode of a JES2 job class (on a job class by job class basis) to MODE=WLM. When that is done, no job in that JES2 job class will be selected by normal JES2 initiators. Instead WLM will be informed of the jobs waiting execution and will start and stop WLM controlled initiators based the current backlog of work.

After conversion of a job's JCL, JES2 places the job into an appropriate class queue to await execution. If the job class is a JES2 managed class (JOBCLASS MODE=JES), JES2 initiators select a job from the class queue (depending on the job's priority) and pass control to MVS to execute your program. If the job class is a WLM managed class (JOBCLASS MODE=WLM), JES provides the WLM with a list of jobs waiting to execute, by service class. Based on installation goals for the service classes, the WLM can determine whether job queue time is a significant delay to a

©Copyright 1994, Computer Management Sciences, Inc.

service class meeting its performance goal. If so, the WLM might start an initiator for a job class assigned to the service class.

Job initiators for WLM managed job classes are controlled dynamically by workload management. These initiators run under the Master Subsystem and are not assigned JES2 job numbers. WLM can adjust the number of initiators on each system based on:

- The queue of jobs awaiting execution in WLM managed classes.
- The performance goals and relative importance of this work.
- The success of meeting these goals.
- The capacity of each system to do more work.

The execution queue delay for the jobs will be included in the calculation of execution velocity or response time for the service classes. The resulting execution velocity or response time will be used to calculate a performance index for the service classes. The resulting performance index will be used by the WLM to assess how well the service classes are meeting their performance goals. If a service class is not meeting its performance goal, the WLM might decide to add an initiator to the job class to attempt to reduce the execution queue delay.

When the WLM examines service class delays, it attempts to manage resource allocation to eliminate the most serious delays to work in the service class, where eliminating the delay would cause performance to be significantly improved. It is important to appreciate that for service class periods with execution velocity goals (and for service class periods with response goals under 20 minutes), the WLM will not examine any individual unit of work. Rather, the WLM will examine delays to all work in the service class period[1].

If jobs in a service class spend most of their active time in a queue waiting for initiation, and if this condition is true for most jobs in the service class, the WLM will consider job initiation as the most serious delay for all work in the service class. Consequently, the WLM will tend to ignore other delays (such as CPU delay, MPL delay, etc.) that occur while the jobs are executing. This would have the effect of the WLM not effectively managing the actual execution of the work in the service class!

Prior to OS/390 Version 2 Release 9, only the SMF Type 26 records contained an indication as to whether a job was assigned a job class with

[1] There is an exception to this statement in that the working set management algorithms might be applied to a specific unit of work.

MODE=JES or MODE=WLM. Since SMF Type 26 records are not often kept in a performance data base (and are not available until after a job has ended) analysis of conflicts between JES-managed and WLM-managed initiators was not feasible. Consequently, CPExpert requested that IBM place an indicator in the SMF Type 30 records so that potential problems with WLM-managed initiators could be analyzed. With OS/390 Version 2 Release 9, IBM created the SMF30WMI indicator as a part of the SMF30PF1 (performance section flag byte) variable.

CPExpert examines the SMF30WMI indicator in SMF Type 30 records to determine whether a job is assigned to a JES-managed or WLM-managed initiator. CPExpert then computes the percent of the overall active time that the job spent waiting in the execution queue.

CPExpert produces Rule WLM082 if any jobs assigned to a job class with MODE=WLM spent more than 75% of their total active time waiting in the execution queue..

The following example illustrates the output from Rule WLM082:

```
RULE WLM082: JOBS MIGHT NOT BE SUITABLE FOR WLM-MANAGED INITIATORS

   The execution queue delay for the below jobs accounted for more than
   75% percent of the total active time of the jobs in this job class.
   Since the execution queue delay is so large, the queue delay will
   dominate the calculation of the Performance Index for the service class.
   Consequently, the Workload Manager might not address other execution-
   related delays because they would not significantly affect the
   performance index.  If this finding is regularly produced, you should
   reevaluate whether this job class should be assigned to WLM-managed
   initiators or whether the specific jobs should be reassigned to a
   different job class.

               JOB           SERVICE                        QUEUE     ACTIVE
   JOB NAME   CLASS  SYSTEM   CLASS  INITIATOR TIME        TIME       TIME
   CQQUERY      R    J90     DISCR   17MAR2000:15:01:09   0:00:12   0:00:14
   CQQUERY      R    J90     DISCR   17MAR2000:15:15:29   0:00:12   0:00:14
   CQCHKPT      R    J90     DISCR   17MAR2000:15:15:34   0:00:12   0:00:14
```

Suggestion: If Rule WLM082 is regularly produced, CPExpert suggests that you consider the following alternatives:

- If the job active time is small and many other jobs execute in the service class with small execution queue delay times relative to their total active time, you probably should ignore this finding. This means that the execution queue delay time of the specific jobs that were identified will not dominate the Workload Manager's resource management considerations for work in the service class.

- Review the job class limits (the XEQCOUNT=MAXIMUM= parameter on the JOBCLASS statement) to determine whether the job class limits

restrict the number of jobs executing. If the job class limits do restrict the number of jobs executing in the job class, you may wish to increase the job class limit. Of course, other considerations must be reviewed to determine why the job class limit was selected.

- Respecify the job class for the identified job(s), to remove the job(s) from the job class that has MODE=WLM specified.

- Please provide Computer Management Sciences with feed-back on other options that you consider. Thanks!

Reference: OS/390 JES2 Initialization and Tuning Guide
Section 2.1.4.13.3: Considerations for WLM Goals Applied to Batch Jobs

z/OS JES2 Initialization and Tuning Guide
Section 2.1.4.13.3: Considerations for WLM Goals Applied to Batch Jobs

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM090: SMF Type 30 Interval Recording not turned on

Finding: CPExpert has detected that a service class missed its performance goal, and experienced significant I/O delays. The **TYPE30_V** guidance variable in USOURCE(GENGUIDE) was set to "Y" to indicate that SMF Type 30 Interval Recording was turned on and the **TYPE30DD** guidance variable in USOURCE(GENGUIDE) was set to "Y" to indicate that the modification to MXG or MICS had been implemented to collect detailed DASD information.

CPExpert determined that a service class missed its performance goal, and CPExpert attempted to analyze SMF Type 30(Interval) data related to the service class. However, CPExpert could not find any interval records for the service class which had missed its performance goal.

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided simply to explain why CPExpert cannot analyze potential DASD delay information for the service class which missed its performance goal.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300:	Service class was delayed for UNKNOWN delay
Rule WLM400:	Server service class was delayed for UNKNOWN delay

Discussion: When CPExpert detects that a service class did not achieve its response goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of UNKNOWN delay is that the service class was delayed because of I/O activity.

CPExpert estimates the UNKNOWN delay which could be attributed to DASD I/O[1] by two basic methods:

- CPExpert computes the average DASD response per I/O **for all DASD devices**. This approach is the default method. This approach provides a general assessment of the typical DASD response. However, the approach suffers from the inability to determine whether the service class missing its performance goal experienced the "typical" DASD response.

[1]The I/O delay could be associated with devices other than DASD, of course. CPExpert analyzes the I/O contribution to UNKNOWN delay using an estimated DASD I/O delay because (1) DASD delay is likely to be smaller than I/O delay to any other type of device and any resulting conclusion is more likely to be robust and (2) most important performance goals will be associated with on-line applications and on-line applications typically reference only DASD.

©Copyright 1994, Computer Management Sciences, Inc.

- CPExpert computes the average DASD response per I/O **for those DASD devices referenced by the service class**. This approach provides an assessment of the typical DASD response just for the DASD devices referenced by the service class.

  In order to apply this approach, CPExpert analyzes DASD information extracted from SMF Type 30(Interval) records using the CPExpert modification to MXG or MICS. If this information is available for the service class missing its performance goal, CPExpert can determine which DASD devices were referenced by the service class. CPExpert can then use the DASD response characteristics of these devices in assessing whether DASD I/O could account for a significant part of the UNKNOWN delay.

  Additionally, CPExpert identifies applications which referenced DASD if DASD delay was felt to be significant.

The second approach is implemented only if the guidance in USOURCE(GENGUIDE) indicated that SMF Type 30(Interval) records were available (%LET TYPE30_V = Y was specified) and the guidance indicated that the CPExpert modification to MXG or MICS had been implemented (%LET TYPE30DD = Y was specified).

Both of the above guidance variables were set to "Y", indicating that SMF Type 30(Interval) records were available and that the CPExpert modification to MXG or MICS had been applied. However, CPExpert could not locate any information in the CPEDASD.TYPE30DD SAS data set for the service class missing its performance goal.

The most likely reasons for this finding are (1) an incorrect specification was made in the USOURCE(GENGUIDE) guidance parameters for the system being analyzed or (2) the service class belongs to a subsystem for which Type 30(Interval) records are not being collected.

The following example illustrates the output from Rule WLM090:

```
RULE WLM090:  SMF TYPE 30 INTERVAL RECORDING NOT TURNED ON

   SMF interval recording was not turned on for the service classes listed
   below.  However these classes missed their service goal, and experienced
   significant I/O delays.  CPExpert cannot analyze DASD I/O delays for the
   DASD referenced by the service classes since the interval recording
   information is not available.

                                 MEASUREMENT INTERVAL WHEN
   SERVICE CLASS     PERIOD      SERVICE CLASS MISSED GOAL
   ST_USER              1        14:00-14:15,01MAR1994
   ST_USER              1        14:15-14:30,01MAR1994
   ST_USER              1        14:30-14:45,01MAR1994
   ST_USER              1        14:45-15:00,01MAR1994
```

Suggestion: CPExpert can provide more detailed information about the causes of delays to service classes if the SMF Type 30(Interval) records are collected for the service class missing its performance goal.

You should examine the USOURCE(GENGUIDE) guidance parameters (TYPE30_V and TYPE30DD) to see whether they have been correctly specified. If the specifications do not match the SMF data available or the system being analyzed, you should correct the specifications.

If the TYPE30_V and TYPE30DD specifications are correct, then you probably have specified SMF parameters such that SMF Type 30(Interval) records are not being collected for the subsystem to which the service class belongs. You should assess whether this additional information if worth the modest system overhead incurred by collecting the SMF Type 30(Interval) records.

- You should determine whether the service class is a Started Task. If so, your installation could execute hundreds of Started Tasks and the overhead may be more significant than you wish to incur.

- If you decide that the system overhead is worth the additional information, change the SMF parameters to collect Type 30(Interval) records for the service class.

Reference: System Management Facilities (SMF)
Chapter 4: Customizing SMF

Your turn:

This manual has described how to use the WLM Component to analyze performance constraints with IBM's Workload Manager.

We would appreciate receiving any comments you have regarding this document (style, content, clarity, etc.), or suggestions for improving the WLM Component (ease of use, new rules, changes to rules, etc.). Please send your comments to:

Don Deese
Computer Management Sciences, Inc.
6076-D Franconia Road
Alexandria, VA 22310
www.cpexpert.com

Comments: