CPExpert™

CICS Component

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

This manual applies to the CICS 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
(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

COPYRIGHT INFORMATION

Preface

In 1968, IBM introduced its Customer Information Control System (CICS) - a system that was designed to allow customers to implement transaction-oriented systems more quickly. CICS has gained almost universal acceptance as a general-purpose transaction management system, and is the production backbone of many major corporations.

During the early years of CICS operation, few performance measuring tools existed and little performance data was available. IBM eventually provided a field-developed tool called the CICS Performance Analyzer II (PA II). This tool was greeted with enthusiasm by the CICS user community and IBM subsequently developed the CICS Monitoring Facility. Eventually, commercially-developed measurement tools appeared, and a spectrum of performance monitoring tools are available today.

However, there initially was little documentation available to guide users in interpreting the results from the tools and making changes to improve CICS performance. IBM responded to this lack of guidance by developing the *CICS Performance Guide*. The *CICS Performance Guides* have been made available for all releases of CICS.

The *CICS Performance Guide* has undergone several revisions as new Releases of CICS have been announced. With each revision, IBM has expanded and improved the guidance, with the result that the *CICS Performance Guide* for CICS Transaction Server for OS/390 is arguably the best document produced by IBM!

Although the *CICS Performance Guide* is an admirable document, users must still laboriously review CICS statistics in order to apply the guidance and improve CICS performance. Reviewing the statistics and determining when to apply changes requires a significant amount of time of the most experienced CICS systems personnel. Unfortunately, these are the very individuals typically with the least amount of time available to spend reviewing statistics.

The CICS Component of CPExpert was designed to help analyze the performance of CICS environments by automating much of the analysis process, identifying potential constraints to improved performance, and suggesting ways to eliminate the constraints. The CICS Component accomplishes this by processing CICS statistics and applying many of the performance analysis techniques contained in IBM's *CICS Performance Guide*.

How to use this manual

This document describes how to use the CICS Component of CPExpert to analyze major constraints to improved CICS performance. The manual is organized into six sections and one appendix.

Section 1 provides an introduction to the CICS Component. This section is organized into three chapters. Most of this section can be reviewed for general information.

- Chapter 1 provides a brief background of CICS performance problems and performance analysis.

- Chapter 2 provides an overview of the CICS Component of CPExpert.

- Chapter 3 describes the sources of data used by the CICS Component to analyze constraints to improved CICS performance.

Section 2 provides information on installing the CICS Component (please follow the instructions contained in the *CPExpert Installation Guide* if CPExpert has not been installed). The instructions in this section should be followed when installing the CICS Component. Chapter 2 is included to illustrate the SAS data sets created by MXG that should be present for the CICS Component to analyze CICS performance.

Section 3 describes how to provide data selection and presentation guidance, and specify analysis guidance to the CICS Component. The instructions in this section will be important at any time the guidance need to be changed. This section is organized into two chapters.

Chapter 1 describes how to specify data selection and presentation guidance variables. The instructions in this chapter will be important if you wish to select specific measurement periods for analysis or analyze a different CICS region, or to alter the format of reports produced by the CICS Component.

Chapter 2 describes how to specify analysis guidance variables to guide the CICS Component in its analysis of CICS performance constraints. The instructions in this chapter will be important if you wish to alter the defaults provided with the CICS Component. You should review this chapter so that you appreciate the defaults and how they are used by CPExpert in analyzing CICS performance.

Section 4 provides information on executing the CICS Component. The instructions in this section should be followed closely when executing the CICS Component. Additionally, Chapter 2 contains a checklist for executing the CICS Component.

Appendix A contains a detailed description of each rule that results in a finding based upon the CICS Component analyzing performance of CICS in your environment. 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 CICCPE Module.

Acknowledgments

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, SystemHouse

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 CICS Component of CPExpert:

Bryant Osborn, Litton Computer Services

Debora Reinert, NORDSTROM

A special acknowledgment is due to the authors of IBM's *CICS Performance Guide* for CICS Transaction Server for OS/390 and for z/OS. These documents, building on previous IBM *CICS Performance Guides*, are arguably the best documents produced by IBM! They are well-written, is very readable, and are a practical guide to tuning CICS. The authors of these publications and the earlier versions of the *CICS Performance Guide* deserve thanks from anyone involved with CICS performance analysis!

[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

Page

Page

Page

Page

Section 4: Executing the CICS Component

Section 5: Using the CICS Component

Appendix A: Description of Rules

Page

Exhibits

Changes

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

- The CICS Component has been updated with new analysis in several areas:

  - Rule CIC109: CICS region is approaching maximum capacity, provides analysis of the quasi-reentrant (QR) TCB approaching a limit that IBM indicates is maximum capacity.

  - Rule CIC275: CICS-DB2 peak TCBs is approaching TCBLIMIT, provides initial analysis of CICS-DB2 Connect performance problems

  - Rule CIC276: CICS-DB2 peak tasks on Pool Ready Queue is high, provides initial analysis of CICS-DB2 Connect performance problems

  - Rule CIC277: CICS-DB2 pool threads in use is approaching limit, provides initial analysis of CICS-DB2 Connect performance problems

  The analysis of CICS-DB2 Connect performance problems will be expanded significantly with Release 13.2, with enhanced documentation of the rules listed above and with new rules added to analyze CICS-DB2 Connection problems[1].

  The following rules were added to analyze performance problems with CICS shared data tables:

  - Rule CIC401: Adds were rejected because shared data table was full

  - Rule CIC402: Records not found in CICS-maintained shared data table

  - *Rule CIC403: Records not found in user-maintained shared data table*

  - Rule CIC405: High data set activity for CICS-maintained shared data table

  - Rule CIC406: VSAM data set might not be good candidate for shared data table

[1] The CICS-DB2 Connect rules listed above were suggested by **Rexaldo Avendano** (Kaiser Permanente). However, his suggestions were made as I began final testing of Release 13.1. These rules were added to Release 13.1, but the documentation is incomplete.

The following rules were added to analyze performance problems with coupling facility data tables (CFDT)[2]:

- Rule CIC420: High percent of structure entries were in use
- Rule CIC421: High percent of structure elements were in use
- Rule CIC422: High percent structure requests encountered length error
- Rule CIC423: High percent entries (CFDT or item) not found
- Rule CIC424: List full condition was reached for CFDT
- Rule CIC425: The CFDT list structure was out of space
- Rule CIC426: High percent version check failed for an entry being updated
- Rule CIC440: LOC=ANY storage request initially failed and was retried
- Rule CIC441: LOC=ANY requests were unable to obtain storage and failed
- Rule CIC442: LOC=ANY low percent minimum free storage
- Rule CIC443: LOC=BELOW storage request initially failed and was retried
- Rule CIC444: LOC=BELOW requests were unable to obtain storage and failed
- Rule CIC445: LOC=BELOW low percent minimum free storage

- Rule CIC446: CFDT pool server storage allocation was less than expected

The following rule was added to allow users of the CICS Component to be informed of CPU time used by CICS work represented by a WLM report class. This new logic allows the CICS Component to associate CICS work with WLM report class information in SMF Type 72 records (MXG file TYPE72GO). The CICS Component will produce findings when designated conditions exist.

- Rule CIC700: CICS report class used more than specified percent CPU time

The initial rule relates only to percent CPU time used by specific CICS work, but future rules could relate to other information contained in the MXG TYPE72GO file.

[2]Thanks to **Rexaldo Avendano** (Kaiser Permanente) for providing test data so I could develop the CFDT rules.

- 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 now specifies OPTIONS COMPRESS=N; to override any site specification for file compression. Experiments have shown that CPExpert code runs significantly faster (using much less CPU time) if compression has been turned off.

CPExpert Release 12.2:

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

- Add support for CICS/TS for z/OS Release 2.2.

- Add code that allows non-standard CICS statistics interval data to be used with a SAS/ITSV (renamed SAS/ITRM) performance data base.

- Add documentation to describe the implications when the number of GETMAINs is too high.

CPExpert Release 12.1:

The main changes to the CICS Component for CPExpert Release 12.1 are:

- Add Shared Temporary Storage (TS) Server analysis. This new code includes 20 new rules to analyze:

- Shared TS Queue Server: coupling facility statistics
- Shared TS Queue Server: buffer pool statistics
- Shared TS Queue Server: storage statistics

This new analysis was suggested by **Paul Gordon** of Bank of America, who not only provided the suggestions, but also (in collaboration with **Bryant Osborn** of Bank of America) provided test data[3] so the new code could be tested.

- Remove the dependancy of the CICS Component on the System Initialization Table (SIT) parameters. With this major revision of the CICS Component, all required SIT parameter information is automatically acquired from data available in standard CICS interval statistics (for example, the SIT MXT value is acquired from the XMCMXT value in the MXG CICXMC data set). This change has the effect of significantly reducing the user effort required to install and execute the CICS Component.

- Improve the listing of rules that are suppressed. A rule is suppressed when (1) the required data variables do not exist in your performance data base, (2) the required data variables are not available for the specific CICS region being analyzed, (3) the rule applies to a CICS feature being evaluated, (4) the rule does not apply to the version of CICS being analyzed, or (5) the rule has been 'turned off' the rule in USOURCE(CICGUIDE). The new report lists each suppressed rule and describes why the rule was suppressed.

- Enhance the output of many CICS rules, by providing a better description of the problems and providing variables to reveal the significance of the analysis.

CPExpert Release 11.2:

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

- Add support for CICS/Transaction Server for z/OS, Version 2.1.

- 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).

[3] It should be mentioned that the test data revealed that IBM's *CICS Performance Guides* have many errors with the descriptions of Shared TS Server variables in SMF Type 110 records. IBM Hursley has kindly agreed to correct these documentation errors.

- Add optional links in CPExpert reports (if the SAS ODS feature is invoked), that link rule output to CPExpert documentation for the rules produced.

- Add an option to create summary output showing all rules that were produced, and listing the CICS regions for which the rules were produced (this new option was suggested by **Paul Gordon** of Bank of America).

- Add an option to create summary output by **system**, to show all rules that were produced and list the CICS regions for which the rules were produced (this new option was suggested by **Paul Gordon** of Bank of America).

- Added a new rule (CIC175) to analyze whether CICS LSR pools were seldom used (this rule was suggested by **Paul Gordon** of Bank of America).

- Added a new rule (CIC176) to analyze whether CICS LSR subpools were seldom used (this rule was suggested by **Paul Gordon** of Bank of America).

- Updated appropriate documentation references for z/OS Version 1 Release 2.

CPExpert Release 11.1:

The main changes to the CICS Component for CPExpert Release 11.1 are documentation corrections and updating references for z/OS Version 1 Release 1.

CPExpert Release 10.2:

The main changes to the CICS Component for CPExpert Release 10.2 are simply documentation corrections.

CPExpert Release 10.1:

The main changes to the CICS Component for CPExpert Release 10.1 are to add new rules to analyze the performance implications of system logger use by CICS. These rules were suggested by **Paul Gordon**, Bank of America.

- Add a new rule to analyze whether CICS log stream structure offloads occurred because the structure was 90% full.

- Add a new rule to analyze whether Interim storage was not efficiently used for CICS log streams.

- Add a new rule to analyze whether local storage buffers not efficiently used for CICS DASD-only log streams.

- Add a new rule to analyze whether the DASD staging data set high threshold was reached for CICS log streams.

- Add a new rule to analyze whether frequent log stream DASD-shifts occurred for CICS log streams.

- Add a new rule to analyze whether specific CICS log streams caused structure to reach high threshold

- Add a new rule to analyze whether specific CICS log stream consumed most of structure resources

CPExpert Release 9.2:

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

- Add a new rule to analyze whether the value of the DSHIPINT value in the System Initialization Table (SIT) might be too large.

- Add a new rule to analyze whether more than one string was specified for VSAM ESDS files that have only write activity.

- Add a new rule to analyze whether VSAM ESDS files have a mixture of read and write activity, but that are *primarily* write-only, and should have two file definitions specified.

- Add a new rule to analyze whether SUBTSKS=1 was specified in the System Initialization Table (this rule was suggested by **Bryant Osborn** of Bank of America).
- Add a new rule to analyze whether CICS waited for a logger "structure full" condition for log streams residing in a coupling facility.

- Add a new rule to analyze whether CICS waited for a logger "staging data set full" condition for DASD-only log streams.

CPExpert Release 9.1:

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

- Add an option to extract only CICS interval statistics files and variables that are required for analysis of CICS performance constraints. This option is suitable for sites with extremely large CICS performance data base.

- Add an option to exclude listing of the SIT and rules that are “turned off” by the CICS Component. This option is suitable for sites that analyze a very large number of CICS regions and otherwise might have an undesirably large output from the CICS Component.

- Add an option to suppress the analysis and findings of particular rules, or suppress the analysis and findings only for particular CICS regions. The desire for this feature 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 CICS region.

- Revise the process for specifying guidance for CPExpert to analyze the TCLASS times at MAXACTIVE.
- Add options to analyze the tasks active in a CICS region, as controlled by the MXT keyword in the SIT.

- Add an option to specify unique analysis guidance for individual CICS regions. This option applies to any user who analyzes multiple CICS regions during a single execution of the CICS Component. The various CICS regions typically provide service to different users, the regions often have different applications, and the regions may even be a mixture of test regions and production regions. One set of analysis guidance variables do not always apply to all regions being analyzed. Consequently, CPExpert now provides the capability to override the basic guidance on a region-by-region basis.

- Add a new rule (suggested by **Paul Gordon** of NationsBank) to analyze whether the Maximum Task (MXT) specification may be too small.

- Add a new rule (suggested by **Paul Gordon** of NationsBank) to analyze whether the Peak EDSA usage is approaching EDSALIM value.

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

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

CPExpert Release 8.2:

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

- Update the Component to support OS/390 Version 2 Release 6.

- Correct code based on errors reported by users.

CPExpert Release 8.1:

The main changes to the CICS Component for CPExpert Release 8.1 are to update the Component to support CICS/Transaction Server, Release 1.2. Additionally, a new rule (suggested by **David Ehresman** of University of Louisville, KY) analyzes the value of the TRTABSZ keyword in the System Initialization Table (SIT).

Section 1: Introduction

This section provides a brief perspective of CICS performance evaluation, provides an overview of the CICS Component of CPExpert, describes the sources of data used by the CICS Component to analyze system performance, and describes the analysis the CICS Component performs.

Chapter 1: Background

From a global view, CICS performance analysis can be viewed from three perspectives: (1) the interaction between a CICS region and the MVS environment, (2) the performance constraints of an individual CICS region, and (3) the performance of tasks executing in the CICS region. CICS performance constraints may arise in each of these areas. However, the CICS Component primarily addresses the second area (the performance constraints of an individual CICS region).

- **The interaction between a CICS region and the MVS environment**. Performance constraints caused by the interaction between a CICS region and the overall MVS environment relate to such areas as:

    - The processor speed, whether the CPU dispatching priority assigned to the CICS region is sufficient high that CICS has adequate processor resources, or whether a sufficiently high performance goal and goal importance has been assigned for the service class to which the CICS region (or CICS transactions) have been classified.

    - The I/O facilities available, and how those facilities are allocated between CICS and other work executing in the system.

    - The swapping and paging controlled by the MVS Auxiliary Storage Manager, and whether the CICS region is affected by the swapping and paging.

    - The communications network, how CICS users access the CICS region through the network, and the communications software parameters controlling the access.

    At this level, it is necessary to determine management objectives for overall performance and resource use of the CICS region relative to that of other workloads (e.g., other CICS regions, TSO transactions, other on-line applications, etc.) executing on the system.

    Management must decide how much of the overall system's resources the CICS region should be allowed, versus the resources required to support other workloads. If the CICS region is given inappropriate resource priorities (processor, DASD, etc.), then CICS performance might not meet expectations, regardless of the amount of tuning.

The WLM Component and the DASD Component of CPExpert, as well as other commercially-available tools, can be used to investigate potential problems in the overall MVS environment.

- **The performance of an individual CICS region**. Constraints to improved performance of an individual CICS region are related to the general controls available for effecting how CICS uses resources, and how CICS distributes access to the resources among tasks attached to the CICS region.

  - DASD and file control facilities, and how CICS parameters are established to minimize the amount of physical I/O operations and optimize I/O access times.

  - Data communications facilities, and how CICS parameters are established to control the use of these facilities.

  - Virtual and real storage, and how the CICS region options have been specified to minimize storage constraints.

  - Overall CICS facilities, and how these facilities are employed by the CICS region.

  These controls are in the System Initialization Table (SIT) and other tables used by CICS. The effect of selecting particular values for the controls generally can be observed from information available in standard CICS interval statistics and information available from monitoring tools (e.g., IBM's CICS Monitoring Facility).

  The CICS Component of CPExpert evaluates constraints to improved performance of an individual CICS region and suggests changes to the controls affecting performance.

  With MVS (Goal Mode) and if CICS Version 4.1 or a higher release of CICS is installed, the WLM Component of CPExpert can be used to investigate constraints to a CICS region from an overall system view.

- **The performance of individual CICS tasks or application systems**. The performance of individual CICS tasks or application systems generally relate to application system design, individual coding strategies, and specific file structures or data base design. Other commercially-available tools can be used to analyze application performance. Consequently, the CICS Component of CPExpert performs minimal analysis in this area.

Chapter 2: The CICS Component of CPExpert

The main purpose of the CICS Component of CPExpert is to evaluate the performance of an individual CICS region, identify potential constraints to improved performance, and suggest ways to eliminate the constraints.

CICS Component was implemented with the basic philosophy that (1) CICS performance analysis should be an ongoing process, (2) changes in response to constraints normally will be made only if the problems are persistent, (3) data required for the performance analysis is available from a variety of sources and exists in a variety of forms, and (4) a tool generally useful for analyzing CICS performance should tailor its analysis based upon available information.

- CICS performance evaluation should be done on a regular basis, with problems or potential problems brought to the attention of CICS systems personnel. If a consistent analysis methodology were automatically applied to the basic CICS performance metrics, performance problems can be identified as they arise without requiring a significant amount of time from CICS systems personnel.

  The CICS Component is designed to be used on a daily basis, analyzing interval statistics normally recorded by CICS.

- Many installations carefully restrict the data elements contained in the performance data base. This management action is done because of the potentially massive amount of data generated by the tools.

  The CICS Component primarily analyzes CICS information contained in CICS/ESA, CICS Transaction Server for OS/390, of CICS Transaction Server for z/OS standard interval statistics recorded in SMF[1] and processed by MXG. CICS information must be available in a performance data base created by MXG for CPExpert's analysis.

- A useful CICS analysis tool should not require an inordinate amount of tailoring by users. Rather, the tool should determine the data elements available and generate analysis appropriate to the available information.

  The CICS Component automatically generates code appropriate to the source of data and performs analysis in each area only if the required performance-related information is available. At the beginning of execution, the CICS Component lists the rules (if any) which will be suppressed based upon data availability.

These features and the analysis are provided by SAS macros and by extensive use of the SAS macro language. These SAS macros (1) determine the data elements present if a performance data base is used, (2) generate coding to shape the available CICS data, (3)

[1] The CICS Component optionally processes system logger information contained in SMF Type 88 records.

generate coding to analyze the CICS statistics depending upon the availability of information and the type of analysis, (4) evaluate the data to assess potential constraints to improved CICS performance, and (5) report the results from the evaluation.

- **Determine data elements in performance data base**. The CICS Component **may** determine the data elements in a MXG performance data base (this action is done only if the STANDARD guidance variable is set to NO in the CICGUIDE module of USOURCE).

  The CICS Component determines which data elements are present so that CPExpert generates only those rules which have data to analyze. The CICS Component lists any rules which are suppressed because of missing data elements or rules which are suppressed because they do not apply to the version of CICS being analyzed.

- **Shape Data**. Based on the data sources and on the data elements available, the CICS Component generates appropriate code to shape the data for analysis. A variety of options are available which can be used to exclude historical data from analysis. For example, a user can exclude weekends or holidays if these periods are not desired in the data analysis and reporting.

- **Evaluate Data**. The evaluation of the CICS performance-related information is accomplished by rules whose purpose is to identify potential constraints to improved CICS performance. The CICS Component invokes rules to analyze the data only if the data required a particular rule is available. The SAS macro language is used to test for the required data elements. Each rule will be invoked only if the necessary data elements are present or if the rule applies to the version of CICS being analyzed.

- **Report Results**. The CICS Component reports the results from the evaluation, producing a detailed narrative associated with each finding. Optionally, the CICS Component can produce output with Hypertext Markup Language (HTML) if the SAS Output Distribution System (ODS) is available[2].

In normal operation, the CICS Component is used on a daily basis to analyze the standard interval statistics recorded by CICS. The CICS Component conducts its analysis guided by the guidance variables contained in CPEXPERT.USOURCE(CICGUIDE). After the analysis is complete, the CICS Component lists the narrative findings resulting from the analysis.

Some findings may apply only to the performance of CICS for the particular day being analyzed, and some recommendations might involve changes potentially having a significant effect on operation of the CICS region. For many potential problems identified by CPExpert and for many alternatives suggested, you should **not** make a change

[2]The SAS ODS is available with SAS Release 8.0.

immediately. Rather, you should wait to see if the analysis results in the same finding and alternatives, or until CPExpert performs an analysis of historical information.

There are some exceptions to this "wait before making changes" advice:

- You feel that the analysis is generally applicable - that is, it is clear that CPExpert has identified a significant problem which will recur and should be immediately addressed.

- The workload processed on the day being analyzed is particularly critical, and the suggested changes are appropriate because the workload has a high management priority.

Each significant finding as a result of the analysis is described in Appendix A of this document. The description summarizes the finding, lists predecessor findings, discusses the rationale for the finding, and suggests alternatives.

- The summary presents a short description of the finding.

- The discussion describes as much as necessary of the operation of CICS as it relates to the particular finding. The purpose of the discussion is to explain the reasoning behind the finding. If appropriate, the discussion might refer you to related discussions in the CICS Component User Manual, or in a User Manual of another CPExpert component (e.g., the User Manual for the WLM Component might be used as a reference to avoid repeating a detailed discussion).

- The suggestions list possible actions that should be considered based on the findings. In many cases, multiple alternatives 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.)

- Much of the discussion and suggestions have been acquired from IBM's *CICS Performance Guide*. Specific references are given to the appropriate pages in each of the relevant *Guides*, or in other documents (if appropriate). Additionally, some findings or discussions are based on information contained in documents available on IBMLINK[3]. Reference is made to the specific IBMLINK document number and document title for rules where the findings or discussions use information contained in IBMLINK.

[3] IBMLINK is a system that provides electronic access to a number of existing IBM software, application, and product support data bases. These data bases normally contain questions to and answers from IBM product developers or support personnel. Access to IBMLINK is acquired thorough the IBM Information Network.

Chapter 3: Data Sources

CPExpert analyzes the performance of an individual CICS region based upon data from two sources:

- **Guidance information**. Guidance information is contained in CPEXPERT.USOURCE(CICGUIDE). This PDS member contains information that allows you to select specific data to be analyzed, to specify how you wish reporting to be done, and to provide guidance to CPExpert with regard to analysis thresholds that may be unique to your environment.

- **Performance data base information**. The CICS Component can analyze measurement information contained in a performance data base. This performance data base must be created from CICS interval statistics, using MXG. The CICS statistics recorded by CICS/ESA, CICS Transaction Server for OS/390, or CICS Transaction Server for z/OS are written to SMF. If these statistics are to be utilized in the analysis, they must be available in a performance data base. CPExpert does not directly process these statistics.

  Please do not confuse the CICS interval statistics with the CICS monitoring data produced by the CICS Monitoring Facility.

  - The *CICS interval statistics* are written at user-defined intervals, with a default of once per 3 hours. These statistics are written as SMF Type 110 records, and do **not** consume much space in the SMF file. The records are relatively small and normally are written infrequently. They have virtually no impact on the size of the SMF file.

    While you can effectively "turn off" the interval statistics by making the interval extremely large, CICS will still write the statistics at the end of day (at midnight) or when the CICS region is shutdown. Thus, you will receive the statistics at least once per 24 hours.

    CPExpert recommends that you record the interval statistics at least once per 3 hours (the default) or more often if you wish to gain additional insight into CICS performance problems.

  - The *CICS Monitoring Facility* data are written based on the setting of the MN, MNPER, MNEXE, and MNEVE keywords in the CICS System Initialization Table. There are three types of CICS Monitoring Facility data: *performance* class data, *exception* class data, and *SYSEVENT* data.

    - The performance class data consist of detailed information at the transaction level. At least one performance class record is written for each CICS transaction.

- The exception class data consist of information on exceptional conditions experienced by a CICS transaction. These exceptional conditions primarily include waits for storage or files.

- The SYSEVENT class data consist of information which is used primarily to record transaction timing data. The SYSEVENT class includes the terminal ID and elapsed time of each CICS transaction.

  The SYSEVENT class data are not applicable with CICS/ESA Version 4.1 if executing under the Workload Manager (MVS Goal Mode), as CICS provides the Workload Manager with detailed information about each transaction.

The CICS Monitoring Facility records can be extremely voluminous and can consume a significant part of the SMF file.

Additionally, collecting and recording the performance and accounting data can require a significant amount of processor resources. The IBM *CICS Performance Guides* estimate that the overhead is likely to be about 5% to 10%, depending upon the workload.

CPExpert does not require and does not use CICS Monitoring Facility data. Many organizations do not collect CICS Monitoring Facility data because of the amount of transaction-level data recorded. Most of the important performance-related data elements previously recorded in the CICS Monitoring Facility file were incorporated in the CICS/ESA, CICS Transaction Server for OS/390, or CICS Transaction Server for z/OS interval statistics file. In fact, the CICS Monitoring Facility records do not contain much of the information which CPExpert uses to analyze performance. Consequently, CPExpert processes only the CICS interval statistics.

Note that MXG is required for CPExpert to comprehensively process the CICS interval statistics. Consequently, MXG should be acquired if you wish to use CPExpert to analyze CICS interval statistics.

Section 2: Installing the CICS Component

Most of the CICS Component will be installed as normal part of installing CPExpert. However, one step is unique to the CICS Component of CPExpert: placing CICS statistics into a performance data base created by MXG.

CICS statistics are accumulated by CICS management programs into CICS system tables. These statistics can be (1) captured and recorded on request (**requested statistics**), (2) captured and recorded automatically at a specified interval (**interval or automatic statistics)**, (3) recorded at normal termination of the CICS system (**shutdown statistics**), and (4) **unsolicited statistics** for dynamically allocated and deallocated resources. The statistics are written to SMF (unless a different destination is specified).

- Requested statistics can be recorded at any time when specified by the operator in the CEMT transaction. The requested statistics are sent to the default destination CSSL unless the operator specifies another destination in the CEMT transaction.
- Interval or automatic statistics can be recorded at intervals defined by the operator in the CEMT transaction.
- The shutdown statistics are recorded at normal CICS termination if an explicit request for all requested statistics had been made then. The shutdown statistics do not include statistics that were recorded during interval or automatic statistics recording or those in which the counters were reset during requested statistics.
- The unsolicited statistics are recorded for dynamically allocated and deallocated resources.

The CICS statistics contain a wealth of information. In fact, most of the performance analysis techniques described in IBM's *CICS Performance Guide* relate to information available from the CICS shutdown statistics.

For best analysis of CICS performance problems or potential problems **interval statistics** should be gathered and made available for CPExpert's analysis. The interval statistics are gathered by CICS during a specified interval. CICS writes the interval statistics to the SMF data set automatically when the specified interval expires, if:

- Statistics recording status was set ON by the **STATRCD** system initialization parameter (and has not subsequently been set OFF by a CEMT or EXEC CICS SET STATISTICS RECORDING command). The default is STATRCD=OFF. The default should be changed to **STATRCD=ON** to collect the interval statistics.
- ON is specified in CEMT SET STATISTICS.
- The RECORDING option of the EXEC CICS SET STATISTICS command is set to ON.

The default recording interval for the CICS interval statistics is to record the statistics every three hours.

Additional procedures must be used to obtain interval statistics if you are using CICS Shared Temporary Storage, Coupling Facility Data Tables, or Named Counters. These CICS facilities are implemented using a server, started in its own region, for each shared temporary storage pool, coupling facility data table, or named counter. The servers produce the interval statistics for the specific CICS facility that they provide. As a part of the start-up procedures for the servers, two statistics parameters can be provided:

- The **STATSOPTIONS**={NONE/SMF/PRINT/BOTH} specifies whether the server is to produce interval statistics, and the destination for the statistics it produces. The default specification is **NONE** which means that the server does not produce any interval statistics. This default must be overridden for statistics to be produced. STATSOPTIONS=SMF (or BOTH) should be specified for the servers to produce interval statistics and write the statistics to the SMF data set.

- The **STATSINTERVAL**={03:00/hh:mm} specifies the statistics collection interval, in the range 1 minute to 24 hours. This parameter should specify the same collection interval as is used for the normal CICS interval statistics (default of 3 hours).

As mentioned in Section 1, MXG is required for CPExpert to comprehensively process the CICS interval statistics[1].

Exhibit 2-1 illustrates the SAS data sets created by MXG that should be present for the CICS Component to analyze CICS performance. Some of the files shown in Exhibit 2-1 apply only to certain releases of CICS, and might not be applicable to the release of CICS operating at your site.

If you do not retain one of the data sets required for your release(s) of CICS, please specify %LET STANDARD=NO; in USOURCE(CICGUIDE) and the CICS Component will automatically determine which files (and which variables) are present in your performance data base. The CICS Component will produce a report listing the rules that cannot be executed because of missing information.

[1] Even if you normally place CICS data into a MICS performance data base, you must use MXG to process the CICS statistics since detailed analysis of CICS performance issues can be accomplished only if the detailed information is available.

MXG DATA SET	MXG DESCRIPTION	NOTES
CICAUTO	CICS AUTOINSTALL GLOBAL	
CICCONMR	CICS ISC/IRC MODE ENTRY	
CICCONSR	CICS ISC/IRC SYSTEM ENTRY	
CICCONSS	CICS ISC CONNECTION SECURITY	
CICDLIT	CICS DL/I LOCAL TOTALS	
CICDQG	CICS TDQUEUE TRANSIENT GLOBAL	
CICDS	CICS DISPATCHER DOMAIN	
CICDTB	CICS DYNAMIC TRANS BACKOUT	
CICFCR	CICS FILE CONTROL SPECIFIC	
CICJCR	CICS JOURNAL CONTROL	N/A with CICS/TS
CICLDR	CICS LOADER DOMAIN PROGRAM SPEC	
CICLGR	CICS LOG MANAGER JOURNAL STATS	N/A with CICS/TS
CICLGS	CICS LOGSTREAM MANAGER STATS	Applicable with CICS/TS
CICLSRFR	CICS LSRPOOL FILE STATS PER FILE	
CICLSRR	CICS LSRPOOL STATS PER LSR POOL	
CICSDG	CICS SYSTEM DUMP GLOBAL	
CICSMDSA	CICS STORAGE MANAGER DSA/EDSA	
CICSMT	CICS STORAGE MANAGER TSK SUBPOOL	
CICTC	CICS TASK CONTROL GLOBAL	N/A with CICS 4.1 and above
CICTCR	CICS TERMINAL CONTROL	
CICTCLR	CICS TCLASS TRANSACTION CLASS	N/A with CICS 4.1 and above
CICTSQ	CICS TSQUEUE TEMPORARY STORAGE	
CICTSR	CICS TRANSACTION STATS	N/A with CICS 4.1 and above
CICVT	CICS VTAM GLOBAL	
CICXMC	CICS TRANSACTION MANAGER TCLASS	Applicable with CICS 4.1 and above
CICXMG	CICS TRANSACTION MANAGER GLOBAL	Applicable with CICS 4.1 and above
CICXMR	CICS TRANSACTION MANAGER TRANS	Applicable with CICS 4.1 and above
CICXQ1	CICS SHARED TS QUEUE SERVER CF	Applicable with CICS/TS
CICXQ2	CICS SHARED TS QUEUE SERVER BUFFERS	Applicable with CICS/TS
CICXQ3	CICS SHARED TS QUEUE SERVER STORAGE	Applicable with CICS/TS
TYPE88	SMF SYSTEM LOGGER DATA	Applicable with CICS/TS

MXG FILES USED BY CICS COMPONENT

EXHIBIT 2-1

Additionally, one further step may be required if you implement the SAS Output Delivery System to create output that is formatted in Hypertext Markup Language (HTML). This step is described in Section 3: **SAS Output Delivery System**. From a practical view, you might wish to postpone this step until you are comfortable with the CICS Component and its output.

The CICS Component provides many processing options (for example, you may select particular time intervals for analysis, specify specific APPLIDs to analyze, provide guidance to the CICS Component's analysis process, etc.). These processing options are not, strictly speaking, a part of the installation process. They may be employed as you refine your analysis of the performance of your CICS regions. These processing options are described in Section 3 of this User Manual.

Section 3: Specifying Guidance Variables

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

CPEXPERT.USOURCE(CICGUIDE) contains variables to establish the overall guidance for the CICS Component. You specify certain guidance variables when you initially use the CICS Component to evaluate a CICS region, and you modify the variables in the CICGUIDE member whenever you wish to change the guidance to CPExpert.

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

- The data selection and presentation variables allow you to select particular time intervals to be analyzed, specify the VTAM application identifier associated with the CICS region to be analyzed, suppress certain listings, etc. Chapter 1 describes the data selection and presentation variables and how they are used.

- The analysis guidance variables allow you to provide guidance to the CICS Component with regard to analysis thresholds that may be unique to your environment. The defaults for the various thresholds may be appropriate for the analysis performed by the CICS Component. However, you may have unique situations (or you may simply disagree with the defaults selected). Chapter 3 describes the analysis guidance variables and their defaults.

  The analysis guidance variables were designed to allow you to tailor the analysis performed by CPExpert. **Do not hesitate to make changes if the defaults for the analysis guidance variables do not meet your needs**.

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 data to be analyzed and how the results are to be presented. This chapter describes these variables and how the variables are used.

Exhibit 3-1 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the data selection and presentation variables.

For most users, the data selection variables rarely will be modified. This is because the CICS Component normally will process a day's data, and the data will be processed after the data has been placed into a performance data base. The daily files will contain the data you wish CPExpert to evaluate.

The data selection variables allow you to identify the data that the CICS Component is to analyze, even though the CICS data might be in a different performance data base than your standard performance data base (for example, your standard data base might be created by MICS, while the CICS interval statistics might be created by MXG). This capability to identify the location of CICS statistics extends down to the individual SAS file containing CICS statistics.

Some users might wish to restrict the analysis to only a specific shift (for example, you might not particularly care about CICS performance at times other than the prime shift). The data selection variables may be used to select specific measurement intervals to analyze (for example, you may notice that CICS performance is particularly poor during some interval and wish analysis focused only on that interval).

The data presentation variables might be modified after you initially execute the CICS Component, but these variables will not normally be modified after you have established the reports to your satisfaction.

```
***********************************************************************;
*             DATA SELECTION AND PRESENTATION VARIABLES               ;
***********************************************************************;
%LET CICLIB  = &PDBLIB ; * SAS LIBRARY CONTAINING CICS DATA           ;
%LET STANDARD= YES     ; * STANDARD MXG DATA SETS ARE USED            ;
%LET CICAUTO = &CICLIB..CICAUTO ; * SPECIFY MXG CICAUTO  SAS LIB.FILE  ;
%LET CICCONMR= &CICLIB..CICCONMR; * SPECIFY MXG CICCONMR SAS LIB.FILE  ;
%LET CICCONSR= &CICLIB..CICCONSR; * SPECIFY MXG CICCONSR SAS LIB.FILE  ;
%LET CICCONSS= &CICLIB..CICCONSS; * SPECIFY MXG CICCONSS SAS LIB.FILE  ;
.
.
.
.
.
%LET CICLGS  = &CICLIB..CICLGS  ; * SPECIFY MXG CICLGS   SAS LIB.FILE  ;
%LET CICLSRFR= &CICLIB..CICLSRFR; * SPECIFY MXG CICLSRFR SAS LIB.FILE  ;
%LET CICLSRR = &CICLIB..CICLSRR ; * SPECIFY MXG CICLSRR  SAS LIB.FILE  ;
%LET CICNQG  = &CICLIB..CICNQG  ; * SPECIFY MXG CICNQG   SAS LIBRARY   ;
%LET CICRMG  = &CICLIB..CICRMG  ; * SPECIFY MXG CICRMG   SAS LIBRARY   ;
%LET CICSDG  = &CICLIB..CICSDG  ; * SPECIFY MXG CICSDG   SAS LIB.FILE  ;
%LET CICSMDSA= &CICLIB..CICSMDSA; * SPECIFY MXG CICSMDSA SAS LIB.FILE  ;
%LET CICTC   = &CICLIB..CICTC   ; * SPECIFY MXG CICTC    SAS LIB.FILE  ;
%LET CICTCR  = &CICLIB..CICTCR  ; * SPECIFY MXG CICTCR   SAS LIB.FILE  ;
%LET CICTCLR = &CICLIB..CICTCLR ; * SPECIFY MXG CICTCLR  SAS LIB.FILE  ;
%LET CICTSQ  = &CICLIB..CICTSQ  ; * SPECIFY MXG CICTSQ   SAS LIB.FILE  ;
%LET CICTSR  = &CICLIB..CICTSR  ; * SPECIFY MXG CICTSR   SAS LIB.FILE  ;
%LET CICVT   = &CICLIB..CICVT   ; * SPECIFY MXG CICVT    SAS LIB.FILE  ;
%LET CICXMC  = &CICLIB..CICXMC  ; * SPECIFY MXG CICXMC   SAS LIB.FILE  ;
%LET CICXMG  = &CICLIB..CICXMG  ; * SPECIFY MXG CICXMG   SAS LIB.FILE  ;
%LET CICXMR  = &CICLIB..CICXMR  ; * SPECIFY MXG CICXMR   SAS LIB.FILE  ;
%LET CICXQ1  = &CICLIB,,CICXQ1  ; * SPECIFY MXG CICXQ1   SAS LIBRARY   ;
%LET CICXQ2  = &CICLIB..CICXQ2  ; * SPECIFY MXG CICXQ2   SAS LIBRARY   ;
%LET CICXQ3  = &CICLIB..CICXQ3  ; * SPECIFY MXG CICXQ3   SAS LIBRARY   ;
%LET EXTRACT = N                ; * EXTRACT CICS FILES/VARIABLES       ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-1

```
%LET CICDATES=01FEB1991; * START DATE FOR DATA ANALYSIS                ;
%LET CICTIMES=08:00:00 ; * START TIME FOR DATA ANALYSIS                ;
%LET CICDATEE=31DEC9999; * END DATE FOR DATA ANALYSIS                  ;
%LET CICTIMEE=16:00:00 ; * END TIME FOR DATA ANALYSIS                  ;
%LET CICDAT2S= 0       ; * DEFAULT SECOND SELECTION DATE - START       ;
%LET CICTIM2S= 0       ; * DEFAULT SECOND SELECTION TIME - START       ;
%LET CICDAT2E= 0       ; * DEFAULT SECOND SELECTION DATE - END         ;
%LET CICTIM2E= 0       ; * DEFAULT SECOND SELECTION TIME - END         ;
%LET SHIFT   = N       ; * START AND END TIMES REFER TO SHIFT          ;
%LET SYSTEM  =*ALL     ; * PROCESS ALL SYSTEMS                         ;
%LET SYSTEMn =system   ; * PROCESS SYSTEMn (n = 1-9)                   ;
%LET ALL_CICS= N       ; * OPTION TO ANALYZE ALL CICS REGIONS          ;
%LET APPLID1 =         ; * APPLID OF THE CICS REGION TO ANALYZE        ;
%LET APPLIDn =         ; * APPLID OF THE CICS REGION TO ANALYZE        ;
%LET MAXAPPL = 99      ; * MAXIMUM CICS REGIONS TO ANALYZE             ;
%LET LISTOFF = Y       ; * OPTION: LIST RULES THAT WERE TURNED OFF     ;
%LET LISTGDE = Y       ; * OPTION: LIST GUIDANCE PROVIDED TO CPEXPERT  ;
%LET CICSUMRY = N      ; * PRODUCE SUMMARY REPORT                      ;
%LET CICS_SYS = N      ; * PRODUCE SUMMARY REPORT BY SYSTEM            ;
%LET SASODS  = N       ; * CONTROLS WHETHER SAS ODS IS USED            ;
%LET PATH    =         ; * PATH FOR ODS OUTPUT                         ;
%LET FRAME   = CICFRAME; * GENERIC ODS FRAME NAME                      ;
%LET CONTENTS= CICSCONT; * GENERIC ODS CONTENTS NAME                   ;
%LET BODY    = CICSBODY; * GENERIC ODS BODY NAME                       ;
%LET LINKPDF =         ; * INSERT HTML LINKS TO PDF FILES              ;
%LET STYLE   =         ; * ODS HTML STYLE OPTION                       ;
%LET PDFODS  = N       ; * CONTROLS WHETHER SAS PDF IS USED            ;
%LET PDFFILE = filename; * DEFINES THE SAS PDF OUTPUT FILE             ;
%LET URL     = N       ; * CONTROLS .HTM IN SAS ODS FRAME OUTPUT       ;
%LET SYS_FRAM=         ; * USE GENERIC ODS FRAME NAME                  ;
%LET SYS_CONT=         ; * USE GENERIC ODS CONTENTS NAME               ;
%LET SYS_BODY=         ; * USE GENERIC ODS BODY NAME                   ;
%LET SYS_PATH=         ; * USE GENERIC ODS PATH                        ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-1 (Continued)

Chapter 1.1: Specifying SAS library containing CICS data

The optional **CICLIB** guidance variable allows you to specify a SAS library containing your CICS performance data. This SAS library can be different from the SAS library containing your normal performance data acquired from SMF, RMF, etc. The default value for the CICLIB guidance variable points to the SAS library identified by the PDBLIB guidance variable in USOURCE(GENGUIDE).

You should alter the default CICLIB guidance variable only if your CICS performance data base is contained in a SAS library different from the normal performance data base. The value specified for CICLIB is used by CPExpert as the DDNAME used to access CICS performance data.

Chapter 1.2: Specifying standard MXG CICS data sets

The optional **STANDARD** guidance variable applies only if you have tailored the variables retained in the CICS files in your MXG performance data base, or if you do not retain all files specified in Exhibit 3-1. Most organizations using MXG process CICS interval statistics using the standard MXG software. Some organizations tailor the software (using standard MXG exits or modifying the code), to drop variables from the CICS data sets created by MXG or to suppress MXG files.

If you have **not** modified MXG to tailor the CICS data sets or to suppress MXG files, you should specify **%LET STANDARD = YES;** to advise CPExpert that MXG has not been modified. CPExpert will not generate the code to determine which CICS data sets and which variables are present in your MXG performance data base (thus saving processing time). The default value for the **STANDARD** variable is "YES", since most organizations do not modify the MXG data collection process.

If you **have** modified MXG to tailor the CICS data sets, you should specify **%LET STANDARD = NO;** to advise CPExpert that MXG has been modified. CPExpert will generate the code to determine which CICS data sets and which variables are present in your MXG performance data base. CPExpert will suppress any rules that depend upon missing data (and CPExpert will advise you that the rules have been suppressed).

Chapter 1.3: Specifying library.file for MXG data sets

The optional **CICAUTO, CICCFS6D, CICCFS7D, CICCFS8D, CICCFS9D, CICCONMR, CICCONSR, CICCONSS, CICDB2GL, CICDB2RE, CICDLIG, CICDLIT, CICDQG, CICDS, CICDTB, CICFCR, CICJCR, CICLDR, CICLGS, CICLSRFR, CICLSRR, CICNQG, CICRMG, CICSDG, CICSMDSA, CICSMT, CICTC, CICTCR, CICTCLR, CICTSQ, CICTSR**, **CICVT**, **CICXMC**, **CICXMG**, **CICXMR, CICXQ1, CICXQ2,** and **CICXQ3** guidance variables allow you to specify a SAS library.file for any MXG CICS data set used by CPExpert.

These SAS libraries.files can be different from the SAS library.file containing any other MXG data.

The default values for these optional MXG guidance variables point to the SAS library identified by the CICLIB guidance variable in USOURCE(CICGUIDE), which in turn normally points to the SAS library identified by the PDBLIB guidance variable in USOURCE(GENGUIDE). You should alter the default guidance variables only if you have any of these CICS files contained in a SAS library different from the normal CICS performance data base. Additionally, you can specify a file name different from the standard MXG file.

Chapter 1.4: EXTRACT variable

Some CPExpert users have an extremely large CICS environment, and the CICS performance staff is unable to allocate easily the DASD space for SAS libraries because of the significant amount of space required. The **EXTRACT** variable was introduced so these users could extract only the APPLIDs that they wished to analyze and to retain only those variables required by the CICS Component.

When **%LET EXTRACT=Y;** is specified, CPExpert extracts only the MXG data sets and only the MXG variables that are necessary for the CICS Component to analyze CICS performance constraints. **Only the APPLIDs specified (as described in the Chapter 1.10: APPLIDn variable) will be selected.**

The **CICEXTR** module in CPEXPERT.SOURCE contains sample code that can be used to extract SAS data sets and variables.

Chapter 1.5: CICDATES and CICTIMES variables

The CICDATES and CICTIMES variables specify the start date and start time, respectively, for the interval the CICS Component is to analyze. These variables (in conjunction with the CICDATEE and CICTIMEE 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 4, 2002, specify:

```
%LET CICDATES = 04MAR2002;  * START DATE FOR DATA ANALYSIS;
%LET CICTIMES = 08:00:00;   * START TIME FOR DATA ANALYSIS;
```


The CICDATES and CICTIMES variables (and the CICDATEE and CICTIMEE variables described below) are not normally altered from their defaults. These variables are used **only** if you wish to analyze a subset of the data contained in your performance data base. Under most conditions, you simply use the defaults and CPExpert will analyze all available CICS statistics.

Chapter 1.6: CICDATEE and CICTIMEE variables

The CICDATEE and CICTIMEE variables specify the end date and end time, respectively, for the interval of SMF data the CICS Component is to analyze. For example, to specify that data selection should end at 17:00:00 on March 8, 2002, specify:

```
%LET CICDATEE = 08MAR2002;  * END DATE FOR DATA ANALYSIS;
%LET CICTIMEE = 17:00:00;   * END TIME FOR DATA ANALYSIS;
```

Chapter 1.7: CICDAT2S and CICTIM2S variables

The CICDAT2S and CICTIM2S variables are optional. These variables specify the start date and start time, respectively, for a second interval of CICS statistics that the CICS Component is to analyze. These variables (in conjunction with the optional CICDAT2E and CICTIM2E variables) allow you to select a second period of data to analyze, in addition to the period specified by the CICDATES/CICTIMES and CICDATEE/CICTIMEE selection variables. For example, to specify that a second period of data selection should start at 20:00:00 on March 4, 2002, specify:

```
%LET CICDAT2S = 04MAR2002; * START DATE FOR DATA ANALYSIS;
%LET CICTIM2S = 20:00:00;  * START TIME FOR DATA ANALYSIS;
```

Chapter 1.8: CICDAT2E and CICTIM2E variables

The CICDAT2E and CICTIM2E variables are optional. These variables specify the end date and end time, respectively, for a second interval of CICS statistics the CICS Component is to analyze. These variables (in conjunction with the optional CICDAT2S and CICTIM2S variables) allow you to select a second period of data to analyze, in addition to the period specified by the CICDATES/CICTIMES and CICDATEE/CICTIMEE selection variables. For example, to specify that a second period of data selection should end at 22:00:00 on March 4, 2002, specify:

```
%LET CICDAT2E = 08MAR2002;     * END DATE FOR DATA ANALYSIS;
%LET CICTIM2E = 22:00:00;      * END TIME FOR DATA ANALYSIS;
```

Chapter 1.9: SHIFT variable

The SHIFT variable is used with the CICDATES, CICTIMES, CICDATEE, and CICTIMEE variables. CPExpert will use the CICDATES and CICTIMES variables to exclude data before the specified date and time. 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. In the example shown above, data would be processed from 08:00:00 on 4 March until 17:00:00 on 8 March.

- 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.

Chapter 1.10: 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.

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 in their performance data base may wish to evaluate only a single system with the parameters specified in this member of CICGUIDE. 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 CICGUIDE parameters. These different evaluations can be accomplished by different executions of the CICS Component[1]. For each execution of the CICS 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 CICGUIDE guidance member would specify the system identification to which the guidance applied.

Please note that if the **ALL_CICS** guidance variable (described below) is specified as %LET ALL_CICS=Y; to specify analysis of all APPLIDs, the SYSTEM specification **takes precedence** (that is, if SYSTEM is specified as %LET SYSTEM=ssss, only APPLIDs for the "ssss" system will be analyzed.

Chapter 1.11: SYSTEMn variable(s)

The SYSTEMn variable(s) 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 one system to analyze, and the SYSTEMn guidance variable(s) can be used to select up to 9 additional systems to analyze. For

[1] A much better approach might be to specify unique guidance for the CICS regions involved, using the APPLID override guidance options described later in this section. The APPLID override guidance allows unique guidance to be specified for any CICS region.

example, if you wish to analyze data from four systems (named SYSA, SYSB, SYSC, AND SYSX) in a single execution of the CICS 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 1.12: Analyzing all CICS regions in PDB (ALL_CICS variable)

The default approach to analyzing CICS performance is to analyze a single CICS region, or to analyze multiple CICS regions **only if you specifically describe the name of each region (using the APPLIDn approach described below)**. This design is the default because analysts often wish to review the performance only of one or more important CICS regions.

The default approach is unwieldy with large CICS environments in which many CICS regions exist, since an analyst must spend an inordinate amount of time preparing guidance information for CPExpert. With a performance data base that may include hundreds of cloned CICS regions, the default approach becomes unworkable. The **ALL_CICS** guidance variable directs the CICS Component to analyze all CICS regions present in a performance data base.

- If the ALL_CICS guidance variable is specified as **%LET ALL_CICS = Y**, CPExpert will analyze all CICS regions with interval statistics contained in the performance data base.

- The default for the ALL_CICS guidance variable is **%LET ALL_CICS = N** (which means that CPExpert should analyze only those regions specified by the APPLIDn guidance variables described below). If this default is used, each region to be analyzed must be identified to the CICS Component of CPExpert, using USOURCE(CICGUIDE), using the APPLIDn approach described below.

Chapter 1.13: CICS region(s) to analyze - APPLIDn variable

The APPLIDn variable (where “n” normally is “1" to “99") is used to specify the VTAM APPLID of one or more CICS regions to be analyzed. The APPLID associated with the CICS region to be analyzed should be entered in the **APPLIDn** guidance variable.

For example, if you wished to analyze a single CICS region named CICSPROD, you would specify **%LET APPLID1 = CICSPROD;** in USOURCE(CICGUIDE). CPExpert uses the APPLID to select data from a performance data base. Data from the performance data base will be rejected unless the APPLID in the data records matches the VTAM APPLID specified for the **APPLIDn** guidance variable(s).

The following provides guidance concerning the APPLIDn guidance variable:

- Unless the ALL_CICS guidance variable is specified as **%LET ALL_CICS=Y;** in USOURCE(CICGUIDE), you must uniquely identify the APPLID of every CICS region you wish CPExpert to analyze.

- The APPLIDn variable is ignored if the ALL_CICS guidance variable is specified as %LET ALL_CICS=Y; in USOURCE(CICGUIDE).

- The APPLIDn guidance variables need not be numerically sequential, nor must every number be specified. This provides flexibility to add or delete CICS regions from analysis without concern that the APPLIDn guidance are in any particular order. WARNING: CPExpert does not check for duplicate APPLIDn variables. The last specified APPLIDn is the operative one.

- You can temporarily "disable" the analysis of one or more CICS regions by simply replacing the "%" character before the LET with an "*" character, thus signifying to SAS that the statement is a comment.

Alternatively (and often a better choice), is to simply specify **%LET ALL_CICS=Y;** in USOURCE(CICGUIDE) and CPExpert will analyze every region that it encounters. This option does, however, add additional processing time and would not be appropriate if you wanted to analyze only a few important regions.

Chapter 1.14: MAXAPPL variable

As mentioned in the above discussion of the APPLIDn variable, the default maximum values for “n” range between 1-99 (that is, APPLID1 to APPLID99 can be specified). The MAXAPPL variable is used to limit the number of CICS regions that will be analyzed by the CICS Component during a single execution. A limit on the number of CICS regions is desired because additional code gets generated by CPExpert for each additional CICS region that is analyzed, and the number of regions to be analyzed acts a control variable in various loops within SAS coding.

Some users of CPExpert wish to analyze more than 99 CICS regions in a single execution of the CICS Component, and they wish to specify unique guidance for a subset of the CICS regions. This increased analysis can be accomplished by changing the MAXAPPL variable to specify the maximum number of CICS regions to be analyzed. For example, to specify that a maximum of 200 CICS regions should be analyzed, specify:

```
%LET MAXAPPL = 200 ; * ANALYZE A MAXIMUM OF 200 CICS REGIONS;
```

WARNING: More than 99 regions can be analyzed concurrently only if you are executing under SAS Release 8 or above. Certain variables (particularly array element names) will be generated with a variable name larger than eight characters. SAS Release 6 does not support variable names longer than eight characters.

Chapter 1.15: LISTOFF and LISTGDE variables

CPExpert normally provides “administrative” listings that describe data and analysis parameters, before the CICS Component produces the results from its analysis.

- CPExpert normally lists any rules that have been “turned off” so uses can appreciate why the related analysis has not been done. CPExpert can “turn off” rules because there is insufficient data to execute the rules, because the rules do not apply to the release of CICS being analyzed, etc. Additionally, users can “turn off’ rules for a variety of reasons.

- CPExpert normally lists the guidance contained in USOURCE(CICGUIDE). This is done because some users dynamically alter guidance based on the system or VTAM APPLID being analyzed.

Some sites use the CICS Component to analyze a very large number of CICS regions, and the analysis is performed each day. A very large amount of output can be generated because of the large number of CICS regions that are analyzed.

There generally is no change with respect to the rules that are “turned off” for any region, and there generally is no change to the guidance provided to the CICS Component. Analysts at these sites do not wish to see a daily listing of the rules that are “turned off” since the same rules generally are suppressed, and they do not wish to see a listing of the values contained in USOURCE(CICGUIDE).

The **LISTOFF** and **LISTGDE** variables allow these sites to significantly reduce the number of lines of output produced by the CICS Component.

- Specifying **%LET LISTOFF=N;** causes CPExpert to suppress the listing of the rules that have been “turned off” for a particular CICS region.

- Specifying **%LET LISTGDE=N;** causes CPExpert to suppress the listing of the guidance contained in USOURCE(CICGUIDE).

Chapter 1.16: Produce summary of rules - CICSUMRY variable

The optional CICSUMRY variable is used to produce a summary of the rules that apply to all CICS regions.

Some CPExpert users have a large number of CICS regions that are analyzed by the CICS Component. These users wish to analyze each CICS region individually, but also wish to have a summary produced that shows each rule and lists the CICS regions to which the rule applies. The CICSUMRY guidance variable can be used to produce such a summary of all findings.

The default for the CICSUMRY is "N", indicating that no summary of rules is produced. You can specify **%LET CICSUMRY = Y**; in USOURCE(CICGUIDE). This specification will cause CPExpert to generate a listing of all rules that were produced and to list the CICS regions for which the rule applies.

The output when CICSUMRY=Y is specified is similar to the output when ALL_CICS is specified. Consequently, the CICSUMRY variable is ignored when ALL_CICS=Y is specified.

Chapter 1.17: Produce summary by system - CICS_SYS variable

The optional CICS_SYS variable is used to produce a summary of the rules by system.

Some CPExpert users have a large number of CICS regions operating on a large number of MVS systems. These users wish to analyze each CICS region individually, but also wish to have a summary produced by system, that shows each rule and lists the CICS regions to which the rule applies. The CICS_SYS guidance variable can be used to produce such a summary of all findings.

The default for the CICS_SYS is "N", indicating that no summary of rules is produced. You can specify **%LET CICS_SYS = Y**; in USOURCE(CICGUIDE). This specification will cause CPExpert to generate a listing of all rules that were produced, summarized by MVS system, and to list the CICS regions for which the rule applies.

Chapter 1.18: 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.

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

Some additional options have been implemented for the CICS Component. These options allow users to have HTML output produced on a system-by-system basis.

Chapter 1.18.1: SAS Output Delivery System - SYS_FRAM variable

The optional SYS_FRAM variable is applicable only if (1) you exercise the SAS ODS features, and (2) you wish to have HTML output produced on a system-by-system basis.

Some CPExpert users have a large number of CICS regions operating on a large number of MVS systems. These users wish to analyze each CICS region individually, but also wish to have a summary produced by system. The summary shows each rule produced by system, and lists the CICS regions to which the rule applies. As described earlier, the CICS_SYS guidance variable can be used to produce such a summary of all findings.

When the SAS ODS feature is used, the "summary by system" information can be appended to the files (PDSE members) identified by the **FRAME, CONTENTS,** and **BODY** guidance variables (as described above). Alternatively, the "summary by system" information can be placed in its **own file** if users producing a "summary by system" wish the summary to be directed to a different location than the normal output produced by the CICS Component.

The SYS_FRAM guidance variable serves two purposes when the SAS ODS feature is used and if CICS_SYS=Y has been specified:

- If the SYS_FRAM variable is blank (null), the frame, contents, and body information related to the "summary by system" is **appended to** the files specified by the **FRAME**, **CONTENTS**, and **BODY** variables, respectively. The FRAME, CONTENTS, and BODY variables were described previously.

- If the SYS_FRAM guidance variable is not blank, the frame, contents, and body information related to the "summary by system" is **written to** the files specified by the **SYS_FRAM**, **SYS_CONT**, and **SYS_BODY** variables[2], respectively.

  In this case, the SYS_FRAM variable identifies the file (PDSE member) that integrates the HTML table of contents and the HTML body file for the "summary by system" report.

[2] The SYS_CONT and SYS_BODY variables are described below.

A table of contents and the body file are displayed when the frame file is opened using a standard web browser.

However, if the SYS_FRAM variable is identical to the FRAME variable and the SYS_PATH variable (described below) is null or is identical to the PATH variable, the frame, contents, and body information related to the "summary by system" is **appended to** the files specified by the **FRAME**, **CONTENTS**, and **BODY** variables, respectively. This prevents "over writing" the contents of the FRAME, CONTENTS, and BODY information with the "summary by system" information.

The default value used by the CICS Component for the SYS_FRAM variable is **null**, indicating that the "summary by system" information should be appended to the file (PDSE member) described by the FRAME, CONTENTS, and BODY variables. You can alter this default by changing the value of the SYS_FRAM variable. This name is where you would point your browser to examine output created by the CICS Component.

Chapter 1.18.2: SAS Output Delivery System - SYS_PATH variable

The SYS_PATH variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a "summary by system basis.

As described earlier, the PATH variable specifies the file name (DDNAME) of the external file where all HTML output produced by the CICS Component normally would be placed. This file specification would be the same as the DDNAME name used when creating the partitioned data set as described above.

You might wish to place the "summary by system" information in a different location than that specified by the PATH variable. In this case, you can use the SYS_PATH variable to specify a different external file for the "summary by system" information.

If the SYS_FRAM variable is non-blank but the SYS_PATH variable is null, CPExpert will use the file name specified for the PATH variable to place the "summary by system" information.

If the SYS_FRAM variable is blank but the SYS_PATH variable is not equal to the PATH variable, CPExpert will use the file name specified by the SYS_PATH variable, and place SYS_FRAM, SYS_CONT, and SYS_BODY HTML information in the location identified by the SYS_PATH variable.

Chapter 1.18.3: SAS Output Delivery System - SYS_CONT variable

The SYS_CONT variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a "summary by system basis.

The SYS_CONT variable identifies the file (PDSE member) that contains a table of contents to the HTML output from the CICS Component for the "summary by system" information. The contents file links to the body file, which holds the actual "summary by system" output from the CICS Component.

If the SYS_CONT variable is blank (null), CPExpert will use the value specified by the CONTENTS variable.

Chapter 1.18.4: SAS Output Delivery System - SYS_BODY variable

The SYS_BODY variable is applicable only if you exercise the SAS ODS features, and if you wish to have HTML output produced on a system-by-system basis.

The SYS_BODY variable identifies the file (PDSE member) that contains the HTML output from the CICS Component.

If the SYS_BODY variable is blank (null), CPExpert will use the value specified by the BODY variable.

Chapter 2: Analysis Guidance Variables

The analysis guidance variables allow you to provide guidance to the CICS Component as CPExpert applies the CICS analysis rules. The CPEXPERT.USOURCE(CICGUIDE) module contains defaults for each guidance variable. These defaults may be appropriate for the analysis performed by the CICS 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 3-2 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables.

```
*********************************************************************;
*                  ANALYSIS GUIDANCE VARIABLES                      ;
*********************************************************************;
%LET PRODTEST = PROD    ; * PRODUCTION OR TEST REGION               ;
%LET GUIDE    = N       ; * OPTION TO PROVIDE INDIVIDUAL GUIDANCE   ;
%LET applid_x = GUIDE   ; * OVERRIDE GUIDANCE EXISTS FOR applid_x   ;
%LET RPTCLASS = N  ;    * CICS REPORT CLASSES ARE TO  BE ANALYZED   ;
%LET RPRTn    = report_class_name; * REPORT CLASS NAME TO ANALYZE   ;
%LET RPTGUIDE = N       ; * OPTION TO PROVIDE INDIVIDUAL GUIDANCE   ;
%LET rpt_class= GUIDE   ; * OVERRIDE GUIDANCE EXISTS FOR rpt_class_x   ;
%LET CICnnn   = OFF     ; * EXAMPLE: TURN OFF CICnnn RULE           ;
%LET MAXTASK  =1        ; * TIMES MXT VALUE REACHED                 ;
%LET AMAXTASK =Y        ; * TEST WHETHER AMXT VALUE REACHED         ;
%LET CMAX1    =1        ; * TIMES CMXT VALUE REACHED - CLASS 01     ;
%LET CMAX2    =1        ; * TIMES CMXT VALUE REACHED - CLASS 02     ;
%LET CMAX3    =1        ; * TIMES CMXT VALUE REACHED - CLASS 03     ;
%LET CMAX4    =1        ; * TIMES CMXT VALUE REACHED - CLASS 04     ;
%LET CMAX5    =1        ; * TIMES CMXT VALUE REACHED - CLASS 05     ;
%LET CMAX6    =1        ; * TIMES CMXT VALUE REACHED - CLASS 06     ;
%LET CMAX7    =1        ; * TIMES CMXT VALUE REACHED - CLASS 07     ;
%LET CMAX8    =1        ; * TIMES CMXT VALUE REACHED - CLASS 08     ;
%LET CMAX9    =1        ; * TIMES CMXT VALUE REACHED - CLASS 09     ;
%LET CMAX10   =1        ; * TIMES CMXT VALUE REACHED - CLASS 10     ;

/* START TCLASS MAXACTIVE GUIDANCE FOR CICS VERSION 4.1 AND CICS/TS
 tclassname=guidance
*/

%LET ALLOCF   =0        ;  * ACCEPTABLE FAILED ALLOCATE REQUESTS      ;
%LET ALLOCQ   =1        ;  * ACCEPTABLE QUEUED ALLOCATE REQUESTS      ;
%LET CPUWARN = 40%      ;  * PCT CPU USE FOR WARNING MESSAGE (CIC700)  ;
%LET DLIWAIT  =0        ;  * NO. TASKS WAITING FOR DL/I THREADS       ;
%LET ENQPOOL  =75       ;  * PERCENT ENQ CONTROL BLOCK POOL USED      ;
%LET FCGETUPD =500      ;  * MINIMUM FILE CONTROL GET UPDATE THRESHOLD ;
%LET GETMAIN  =25       ;  * NUMBER OF GETMAINS PER TASK              ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-2

```
%LET JCBUFUL  =0         ;  * USER JOURNAL 'BUFFER FULL' CONDITIONS     ;
%LET LSRHITD  =40        ;  * LOOK-ASIDE HIT RATIO, DATA BUFFERS        ;
%LET LSRHITI  =80        ;  * LOOK-ASIDE HIT RATIO, INDEX BUFFERS       ;
%LET LSRINACT =1000      ; * HITS LESS THAN THIS INDICATES INACTIVE POOL;
%LET LSRIOREQ =75        ; * MINIMUM PERCENT I/O REQUESTS LSR FILES     ;
%LET LSRUSE   =100       ; * HITS THAT INDICATES POOL IS SELDOM USED    ;
%LET LSRUSE1  =&LSRUSE   ; * HITS THAT INDICATES POOL 1 IS SELDOM USED  ;
%LET LSRUSE2  =&LSRUSE   ; * HITS THAT INDICATES POOL 2 IS SELDOM USED  ;
%LET LSRUSE3  =&LSRUSE   ; * HITS THAT INDICATES POOL 3 IS SELDOM USED  ;
%LET LSRUSE4  =&LSRUSE   ; * HITS THAT INDICATES POOL 4 IS SELDOM USED  ;
%LET LSRUSE5  =&LSRUSE   ; * HITS THAT INDICATES POOL 5 IS SELDOM USED  ;
%LET LSRUSE6  =&LSRUSE   ; * HITS THAT INDICATES POOL 6 IS SELDOM USED  ;
%LET LSRUSE7  =&LSRUSE   ; * HITS THAT INDICATES POOL 7 IS SELDOM USED  ;
%LET LSRUSE8  =&LSRUSE   ; * HITS THAT INDICATES POOL 8 IS SELDOM USED  ;
%LET NONUSRBF =0         ; * NON-USER INITIATED BUFFER WRITES           ;
%LET NONUSRBF =0         ; * NON-USER INITIATED BUFFER WRITES           ;
%LET PCTD2TCB =80        ; * PCT NUMBER OF TCBS APPROACHING TCB LIMIT   ;
%LET PCTD2THR =80        ; * PCT NUMBER OF THREADS APPROACHING LIMIT    ;
%LET PCTEDSA  =75%       ; * PCT EDSA USED BY PEAK EDSA USAGE           ;
%LET PCTMXTHI =100%      ; * PCT MXT (ACTIVE TASKS FOR CIC108)          ;
%LET PCTMXTLO =75%       ; * PCT MXT (ACTIVE TASKS FOR CIC104)          ;
%LET PCTQRTCB =60%       ; * PCT QR TCB CPU TIME FOR RULE CIC109        ;
%LET PSBWAIT  =0         ; * NUMBER OF WAITS FOR PSB POOL BUFFERS       ;
%LET PVCOUNT  =0         ; * ACCEPTABLE NUMBER OF PV TIMEOUTS           ;
%LET RAPOOL   =0         ; * VTAM REACHED MAX RAPOOL VALUE              ;
%LET SNTCOUNT =0         ; * ACCEPTABLE NUMBER OF SNT TIMEOUTS          ;
%LET STORDUMP =1         ; * STORAGE DUMP GUIDANCE VARIABLE             ;
%LET STRWAIT  =5         ; * PERCENT I/O WAITING FOR VSAM STRINGS       ;
%LET TRANSERR =0         ; * TRANSACTION ERROR GUIDANCE VARIABLE        ;
%LET TRMSKDEL =50        ; * ACCEPTABLE SHIPPED TERMINAL DELETES        ;
%LET TSIOWAIT =5         ; * TEMP STORAGE I/O WAITING FOR BUFFERS       ;
```

SAMPLE DISPLAY OF CPEXPERT.USOURCE(CICGUIDE) MODULE

EXHIBIT 3-2 (Continued)

Chapter 2.1: Specifying Production or Test Region

The **PRODTEST** variable tells the CICS Component whether the CICS region is a production region or is a test region. The analysis performed by the CICS Component varies, depending upon whether the CICS region is a production or test. Some situations may be acceptable in a CICS test region, but would be unacceptable in a CICS production region.

For example, storage violations may be acceptable (and even expected) in a CICS test region. However, a storage violation in a CICS production region should be cause for immediate action. The CICS Component would essentially ignore any storage violations if the CICS region being analyzed were a test region. However, any storage violation detected in a production CICS region would generate a strong recommendation.

Specify **%LET PRODTEST=TEST;** to change the guidance from the default CICS production region classification to a CICS test region classification.

You may wish CPExpert to analyze both production and test regions during a single execution. In this case, you should specify **%LET PRODTEST=PROD** (the default), to ensure that the production regions are properly analyzed. Some of the results from analyzing the CICS test regions may not be relevant (since CPExpert will be analyzing the test regions as though they were production regions, and thus apply more stringent criteria). You can simply discount these results.

Chapter 2.2: Specifying guidance for regions - GUIDE variable

The **GUIDE** guidance variable specifies whether override guidance exists for one or more CICS regions. Please see Section 3 (Chapter 3) for a detailed description of how to specify override guidance for one or more CICS regions.

Chapter 2.3: Identifying regions with guidance - applid_x=GUIDE

The **applid_x=GUIDE** guidance identifies specific regions that have unique guidance. Please see Section 3 (Chapter 3) for a detailed description of how to specify override guidance for one or more CICS regions.

Chapter 2.4: Analyze CICS report classes - RPTCLASS variable

The **RPTCLASS** guidance variable acts as a “switch” to specify whether the CICS Component should analyze CICS Report Classes, based on information contained in RMF Type70(series) records.

Some users of the CICS Component want CPExpert to provide a “warning” about certain system-level conditions related to CICS workload. To meet this requirement, the CICS Component has been enhanced to provide an ability to associate CICS workload with system-level information contained in RMF Type70(series) records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU. For this requirement, logic was implemented to associate CICS workload with the TYPE72GO information. Guidance was implemented using the CPUWARN guidance variable. While other features may be implemented, the “CPU warning” construct will be used to discuss and illustrate both the concept and the implementation of the enhancement.

The “CPU warning” approach requires that a Report Class describing the CICS workload of interest be defined to the Workload Manger (WLM), using the standard WLM workload classification panel. The System Resources Manager (SRM) acquires information related to the defined Report Class, and RMF writes the Report Class information to SMF as part of the SMF Type 72 records. MXG processes the SMF Type 72 records and places information into various files in the MXG performance data base. CPExpert processes the MXG TYPE72GO file to extract information about the defined Report Class. CPExpert produces a “CPU warning” when the defined Report Class uses more than nn% CPU during any RMF reporting interval.

A CPExpert user implements the “CPU warning” feature by the following steps:

- **Define Report Class**. The “Defining Report Classes” section of IBM’s *MVS Planning: Workload Management* document describes how to define report classes to the WLM. In brief, the WLM’s classification rules can assign incoming work to a report class. Report classes can be used to report on a subset of transactions running in a single service class, or can report on a combination of transactions running in different service classes within one report class.

- **Specify that report classes are to be analyzed by the CICS Component**. The **RPTCLASS** guidance variable is used to tell CPExpert that report classes are to be analyzed. If you wish the CICS Component to analyze report classes, specify **%LET RPTCLASS=Y;** in USOURCE(CICGUIDE).

- **Identify the report classes to be analyzed**. You must tell CPExpert which report classes are to be analyzed. This is done by specifying %LET RPTn=report_class_name; in USOURCE(CICGUIDE). For example, to tell CPExpert that the CICSAOR1 report class should be analyzed, specify **%LET RPT1=CICSAOR1;** in USOURCE(CICGUIDE). You can specify up to 99 report classes to be analyzed.

Once the above process has been implemented, the CICS Component will extract appropriate information from the MXG TYPE72GO file, compare the information with the guidance provided, and produce rules in the CIC700(series) if the guidance has been exceeded.

Chapter 2.5: CICS report classes to analyze - RPRTn variable

The **RPRTn** (where “n” normally is “1" to “99") guidance identifies CICS Workload Manager report classes that should be analyzed by the CICS Component. The report class name associated with the CICS workload to be analyzed should be entered in the **RPRTn** guidance variable. Please see Section 3 (Chapter 4) for a detailed description of how to analyze CICS workload in one or more CICS report classes.

For example, if you wished to analyze a single CICS report class named CICSAOR3, you would specify **%LET RPRT1 = CICSAOR3;** in USOURCE(CICGUIDE). In this example, CPExpert uses the RPRT1 guidance variable to select information for the CICSAOR Report Class from the MXG TYPE72GO file.

The following provides guidance concerning the RPRTn guidance variable:

- You must uniquely identify the report class name of every CICS report class you wish CPExpert to analyze.

- The RPRTn guidance variables need not be numerically sequential, nor must every number be specified. This provides flexibility to add or delete CICS report classes from analysis without concern that the RPRTn guidance variables are in any particular order.

- You can temporarily "disable" the analysis of one or more CICS report classes by simply replacing the "%" character before the LET with an "*" character, thus signifying to SAS that the statement is a comment.

Chapter 2.6: Specifying guidance for CICS report classes - RPTGUIDE variable

The **RPTGUIDE** guidance variable specifies whether override guidance exists for one or more CICS report classes. Please see Section 3 (Chapter 4) for a detailed description of how to specify override guidance for one or more CICS report classes.

Chapter 2.7: Identifying CICS report classes with guidance - report_class_x=GUIDE

The **report_class_x=GUIDE** guidance identifies specific CICS report classes that have unique guidance. Please see Section 3 (Chapter 4) for a detailed description of how to specify override guidance for one or more CICS report classes.

Chapter 2.8: Turning OFF CICS Component Rules

The default guidance values for the CICS Component are specified based on either IBM's guidance contained in IBM's *CICS Performance Guides*, 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 CICS Component wish to suppress the analysis and findings of particular rules, or suppress the analysis and findings only for particular CICS regions. 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 CICS region.

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

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

Rules can be turned OFF by specifying **%LET CICnnn = 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(CICGUIDE).

For example, you can turn OFF Rule CIC104 by specifying **%LET CIC104=OFF;** in the *ANALYSIS GUIDANCE SECTION* of USOURCE(CICGUIDE).

If you are analyzing multiple CICS APPLIDs during a single execution of the CICS Component and you are specifying specific guidance for particular CICS regions, you can turn rules OFF (or turn rules ON) for particular regions. For example, you may wish to turn OFF a rule normally, but wish to have that rule ON for a particular CICS region. Simply specify **%LET CICnnn=OFF;** in USOURCE(CICGUIDE), where "nnn" is the rule you wish to turn OFF. Then you can specify **%LET CICnnn=ON;** in the USOURCE(applid_x) member applying to the particular CICS region for which the rule should be ON.

Please note that the CICS Component 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.

Chapter 2.9: Times at MXT - MAXTASK variable

The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Beginning with CICS Release 1.7, CICS will not create a Task Control Area (TCA) for a new task when the number of tasks concurrently in the region reaches the MXT value. The MXT value (from the SIT or as modified with operator overrides) is contained in variable XMGMXT in the MXG file CICXMG.

CPExpert produces Rule CIC101 if the CICS interval statistics reported that the maximum tasks value (contained in variable XMGMXT) was reached more than the **MAXTASK** guidance variable and CPExpert did not detect a storage constraint.

The default specification for the guidance variable is **%LET MAXTASK = 1**. This value indicates that CPExpert will produce Rule CIC101 if the CICS region reached MXT more than once. You can provide different guidance to CPExpert by changing the MAXTASK if you feel that Rule CIC101 is produced spuriously.

Chapter 2.10: Times at AMXT - AMAXTASK variable

The AMXT operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks in the CICS region. All new and resumed tasks must pass the AMXT limit before CICS dispatches the tasks. CICS marks all new and resumed tasks "nondispatchable for AMXT reasons." The AMXT value (from the SIT or as modified with operator overrides) is contained in variable DSGAMXT in the MXG file CICDS.

CPExpert produces Rule CIC102 if the CICS statistics reported that the AMXT value (contained in variable DSGAMXT) was reached and CPExpert did not detect a storage constraint.

The default specification for the guidance variable is **%LET AMAXTASK = Y**. This specification indicates that CPExpert will produce Rule CIC102 if the number of active tasks in the CICS region was as high as the AMXT value. You can "turn off" this rule by specifying **%LET AMAXTASK = N;** if you feel that Rule CIC102 is produced prematurely.

Chapter 2.11: Times at CMXT - CMAXn variables

This guidance variable is applicable only for CICS Releases prior to CICS/ESA 4.1 (beginning with CICS Release CICS/ESA 4.1, the MAXACTIVE attribute for a transaction class is used).

The Maximum Task Class (CMXT) operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks associated with particular task classes. Up to 10 unique task classes can be defined, and a maximum number of active tasks is specified for each class. The CMXT value (from the SIT or as modified with operator overrides) for each task class is contained in variable A15MXTM in the MXG file CICTCLR.

CPExpert provides guidance variables for each task class. These guidance variables are CMAX1 through CMAX10, corresponding to Task Class 1 through Task Class 10, respectively.

CPExpert produces Rule CIC103 if the CICS statistics reported that the maximum active tasks value (contained in variable A15MXTM) for any class was reached more than the CMXT guidance value for the class, and if the maximum number of concurrently attached tasks for the task class was greater than the CMXT specification for the task class.

The default specification for the guidance variables is **%LET CMAXn = 1**, where "n" is the appropriate class task. This value indicates that CPExpert will produce Rule CIC103 if the CICS region reached CMXT limit more than once for the particular class, and if additional tasks were attached after the limit was reached. You can provide different guidance to CPExpert by changing the CMAXn variables if you feel that Rule CIC103 is produced prematurely.

Chapter 2.12: Times at MAXACTIVE - *tclassname* variables

This guidance variable applies with CICS/ESA Release 4.1, CICS Transaction Server for OS/390, or CICS Transaction Server for z/OS.

The MAXACTIVE attribute for a transaction class can be used to control the number of active tasks in the transaction class. The MAXACTIVE value (from the SIT or as modified with operator overrides) for each transaction class is contained in variable XMCMXT in the MXG file CICXMC.

The MAXACTIVE attribute limits the number of transactions for a specific transaction class, while the MXT value (specified in the System Initialization Table) limits the total number of transactions in the CICS region. Please refer to Rule CIC105 for additional discussion of the MAXACTIVE attribute.

Prior to CICS/ESA Release 4.1, the CMXT values were used to limit the number of active tasks in a transaction class, and up to 10 transaction tasks could be defined (numbered 1-10). With CICS/ESA Release 4.1, a transaction class is given a **name**, rather than a **number**. Consequently, the guidance to CPExpert is different with CICS/ESA Release 4.1.

CPExpert produces Rule CIC105 if the CICS interval statistics reported that the MAXACTIVE value (contained in variable XMCMXT) for any class was reached more than the guidance value for the class. The default for the guidance value is "1", indicating that Rule CIC105 will be produced if the MAXACTIVE value was reached more than one time. This default may cause Rule CIC105 to be produced for a transaction class when you have explicitly constrained the maximum number of tasks in a transaction class. Consequently, you may wish to tell CPExpert not to produce Rule CIC105 unless the MAXACTIVE value is exceeded a much larger number of times.

Initially, CPExpert provided guidance variables for each transaction class, using the **transaction class name** as the guidance variable. For example, suppose that "CLASSA" had been specified as the name for a transaction class. The guidance variable to CPExpert would have been specified as **%LET CLASSA = n** where "n" is the guidance for CLASSA. This process created SAS macro variables as the TCLASS name, and the value of the macro variable was the guidance. CPExpert would produce Rule CIC105 if the MAXACTIVE value was reached more than "n" times for the transaction class. Sadly, this approach did not always work.

CICS transaction class names can have special characters “@, #, and $” in the TCLASS name. Unfortunately, SAS will not accept these special characters imbedded in a SAS macro variable name. Consequently, a different approach was designed.

Exhibit 3-3 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables. As illustrated in Exhibit 3-3, guidance for the TCLASS MAXACTIVE is specified **inside** the SAS macro comment statements (/* and */). The specification is “tclassname = guidance” where “tclassname” is the name of the transaction class and the “guidance” is the number of times that the transaction is allowed to reach MAXTASK before Rule CIC105 is produced. For example:

```
/* START TCLASS MAXACTIVE GUIDANCE
   $067 = 50
*/
```

would be used to specify that CPExpert should produce Rule CIC105 when the $067 transaction class reached MAXACTIVE more than 50 times during a CICS interval statistics period.

Chapter 2.13: Failed ALLOCATE Requests - ALLOCF variable

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. A part of the communication link is the allocation of "sessions" which can send or receive requests. Multiple sessions can be active on the communication link, and the sessions are defined to CICS as "send" sessions or "receive" sessions. (Please refer to Rule CIC260 for more discussion of communication links and sessions.)

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command. With LU6.2, ALLOCATE requests can be for **any** modegroup defined with the connection (a **generic** allocation request), or may be for a **particular** modegroup defined with a connection (a **specific** allocation request).

CICS will process a session ALLOCATE request only if the communication link is established and operational between the two systems. CICS will reject the ALLOCATE request if the connection has been released, the connection is out of service, or the mode group has been closed.

CPExpert produces Rule CIC265 if the number of Failed Link Allocates (A14ESTAF) is greater than the **ALLOCF** guidance variable, for requests to allocate **generic** sessions. CPExpert produces Rule CIC266 if the number of Failed Link Allocates (A20ESTAF) is greater than the **ALLOCF** guidance variable, for requests to allocate **specific** sessions with a modegroup.

The default specification for the ALLOCF guidance variable is **%LET ALLOCF = 0**, indicating that no failed allocation requests are acceptable. You can provide different guidance to CPExpert by changing the ALLOCF guidance variable if you feel that Rule CIC265 or Rule CIC266 are produced prematurely.

Chapter 2.14: Unsuccessful ALLOCATEs - ALLOCQ variable

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. A part of the communication link is the allocation of "sessions" which can send or receive requests. Multiple sessions can be active on the communication link, and the sessions are defined to CICS as "send" sessions or "receive" sessions. (Please refer to Rule CIC260 for more discussion of communication links and sessions.)

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command. With LU6.2, ALLOCATE requests can be for **any** modegroup defined with the connection (a **generic** allocation request), or may be for a **particular** modegroup defined with a connection (a **specific** allocation request).

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available or until the transaction "times out" if the DTIMOUT parameter is used in DFHPCT. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

Some queuing for allocation requests may be unavoidable because an installation may have deliberately restricted the number of sessions to minimize resource use by CICS. Similarly, some return of session ALLOCATE requests to the transaction because sessions were unavailable (if the NOQUEUE option were selected) may be unavoidable for the same reasons.

However, queuing of session ALLOCATE requests delays transactions and causes resources held by the transaction to be unavailable for other work. If the NOQUEUE option were selected, the transaction may implement alternative processing or simply delay the ALLOCATE request for a time interval in hopes that a session will become available. The NOQUEUE option may have less impact, depending upon the specifics of the application. In either case, it is not normally desirable to have ALLOCATE requests unsatisfied because sessions are unavailable.

CPExpert detects a **potential** problem with unavailable sessions in a variety of situations. These situations are described by rules in the CIC260(series). Most of the final decisions made by rules in the CIC260(series) compare values against the **ALLOCQ** guidance variable. The default specification for this variable is **%LET ALLOCQ = 1**, indicating that the CIC260(series) rules would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available. **This low default value is intended only to alert you to a potential problem with the number of sessions defined, and is intended to make you aware of this analysis mechanism**.

For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** guidance variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests queued. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few queued requests, but are concerned about the situations when hundreds or thousands of requests are queued.

Chapter 2.15: CICS workload used excess CPU - CPUWARN variable

Some users of the CICS Component want CPExpert to provide a “warning” about certain system-level conditions related to CICS workload. To meet this requirement, the CICS Component has been enhanced to provide an ability to associate CICS workload (as identified by Workload Manager report classes) with system-level information contained in RMF Type 72 records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU. For this requirement, logic was implemented to associate CICS workload with the TYPE72GO information. Guidance was implemented using the **CPUWARN** guidance variable.

CPExpert produces Rule CIC700 if the CICS workload identified by a report class used more than the percent CPU identified by the **CPUWARN** guidance variable. The default specification for the CPUWARN guidance variable is **%LET CPUWARN = 40%**. You can provide different guidance to CPExpert by changing the CPUWARN variable.

Chapter 2.16: Tasks waiting for DL/I threads - DLIWAIT variable

The DLTHRED operand in the System Initialization Table (SIT) specifies the number of concurrent DL/I threads that can be allocated for IMS/VS data bases. The number of concurrent DL/I threads limits the number of tasks concurrently scheduled for use of IMS/VS resources.

A significant amount of real and virtual storage can be required if a large number of DL/I threads are specified. However, if the number of DL/I threads is too low, tasks can wait for threads. If tasks wait for DL/I threads, they tie up storage while they are waiting, and response suffers. Therefore, there is a tradeoff between the amount of storage allocated for DL/I threads and the delays caused by tasks waiting for threads.

The CICS interval statistics report the number of times tasks waited for DL/I Threads. CPExpert produces Rule CIC190 if storage was not a constraint, and if the number of times tasks waited for DL/I Threads was greater than the **DLIWAIT** guidance value. The default specification for this guidance variable is **%LET DLIWAIT = 0**. You can provide different guidance to CPExpert by changing the DLIWAIT variable.

Chapter 2.17: IMS ENQ pool space used - ENQPOOL variable

The ENQPL operand in the SIT specifies the number of blocks in the IMS enqueue control block pool. The enqueue control block pool is used only if program-isolation scheduling is being used. If the ENQPL value is too small, the IMS/VS DB task abends with a U0775 pseudoabend message, causing dynamic backout of the changes.

Obviously, if the IMS/VS DB task abends, action would be taken to increase the ENQPL value. However, CPExpert analyzes CICS statistics to give an advance warning that such abends may occur with the present setting of the ENQPL operand.

CPExpert produces Rule CIC193 if the percent of the ENQ control block pool used was greater than the **ENQPOOL** guidance variable. The default specification for this guidance variable is **%LET ENQPOOL = 75**, indicating that Rule CIC193 will be produced when more than 75% of the ENQ control block pool is used. You can provide different guidance to CPExpert by changing the ENQPOOL variable if you feel that Rule CIC193 is produced prematurely.

Chapter 2.18: Percent unnecessary UPDATE - FCPCTUPD variable

A CICS application accesses VSAM data sets using CICS file control commands. The file control commands can optionally specify that an update is to occur with the record being read (keyword UPDATE). The UPDATE option guarantees read integrity for the record. One result of this UPDATE option is that the record is locked (and, depending on the type of file, the control interval is locked). Another result of this UPDATE option is that, if the VSAM data set is assigned to a CICS-maintained data table, the VSAM source data set must be referenced by VSAM before the record is referenced in the data table reference.

Both results from using the UPDATE option cause overhead and potentially degrade performance. Consequently, the UPDATE option should be used only if the record is actually updated or it is deleted.

File control statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed a VSAM data set using the UPDATE option but did not subsequently update or delete the records.
CPExpert produces Rule CIC177 when the percent unnecessary UPDATE option is more than the value specified by the PCTFCUPD guidance variable in USOURCE(CICGUIDE). The default value for the PCTFCUPD is 25 indicating that CPExpert should produce Rule CIC177 whenever more than 25% of the VSAM file accesses with the UPDATE option did not result in a corresponding change to the VSAM source data set.

CPExpert normally suppresses this finding if less than 500 GET UPDATE file control commands were issued against the file. You can specify a different threshold for suppressing Rule CIC177 by altering the **FCGETUPD** guidance variable in USOURCE(CICGUIDE).

Chapter 2.19: Number of GETMAINs - GETMAIN variable

The CICS interval statistics report the number of times CICS acquired storage for a task (CICS issued a GETMAIN). The number of GETMAINs is related to the design and use of the tasks and applications. Consequently, there are no CICS tuning controls relating to the number of GETMAINs.

GETMAINs impose system overhead for the system code necessary to execute the GETMAIN. It is desirable to minimize the GETMAINs per task to minimize this overhead, although tasks obviously must issue GETMAINs to acquire any necessary storage.

CPExpert produces Rule CIC110 if the result is greater than the **GETMAIN** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET GETMAIN = 25**. As a "rule of thumb" indicated in IBM documents, there should be fewer than an average of 25 GETMAINs per task. More than this value generally indicates that the task may be improperly designed or that it is experiencing performance problems of some type. You can provide different guidance to CPExpert by changing the GETMAIN variable if you feel that Rule CIC110 is produced prematurely.

Chapter 2.20: "Buffer full" for user journals - JCBUFUL variable

A user journal may be used to record file control or message activity during CICS execution. CICS can be directed to perform automatic journaling by specifying entries in either the file control table or the program control, or users can provide explicit user journaling routines.

For systems prior to CICS Release 3.2.1, a single buffer is acquired for a user journal (CICS Release 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

If the buffer is full, tasks using the user journal must wait. This wait time delays the tasks and lengthens response. Additionally, there is potentially a more serious effect for the overall CICS region when tasks wait for user journal, if the journal is used by several tasks. When tasks wait for a journal output, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPExpert produces Rule CIC221 if the number of times a buffer full" condition exceeded the CPExpert **JCBUFUL** guidance variable. This guidance variable is provided so that the rule will not be produced spuriously in those situations when system design decisions require user journals to be written frequently and the BUFSUV operand is insufficient to meet the requirement.

The default specification for this variable is **%LET JCBUFUL = 0**, indicating that the buffer full condition should not be encountered for user journal logs. You can provide different guidance to CPExpert by changing the JCBUFUL variable if you feel that Rule CIC221 is produced prematurely.

Chapter 2.21: Look-aside read hit - LSRHITD and LSRHITI variables

VSAM files assigned to a Local Shared Resources (LSR) pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations.

There is an extremely important advantage to using LSR pools for VSAM files: VSAM will use its "look-aside" logic to determine whether a required control interval (CI) is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

The CICS interval statistics provide information about the number of times an I/O request was satisfied because VSAM found the data in a LSR buffer. CPExpert calculates the percent of "look-aside hits" versus total I/O access operations (the sum of "look-aside hits" and "buffer reads"). The resulting percent is compared with one of two guidance variables, depending upon how the subpool is used.

- If the subpool is used for data records, the percent is compared to the **LSRHITD** guidance variable.

- If the subpool is used for index records, the percent is compared to the **LSRHITI** guidance variable.

The default values for the LSRHITD and LSRHITI guidance variables are 40% and 80%, respectively. If the subpool is used for both data and index records, CPExpert tests the index **LSRHITI** guidance variable first. This is done since index records should have a higher hit ratio than data records.

CPExpert produces Rule CIC165 if the percent of "look-aside hits" is lower than the appropriate guidance variable. The default specification for the guidance variables is **%LET LSRHITD = 40** (for data records) and **%LET LSRHITI = 80** (for index records). You can provide different guidance to CPExpert by changing the LSRHITD and LSRHITI variables.

Chapter 2.22: Inactive pool - LSRINACT variable

As described in the above discussion related to LSRHITI and LSRHITD, CPExpert produces Rule CIC165 if the percent of "look-aside hits" is lower than the appropriate guidance variable. However, some LSR pools are seldom used, and Rule CIC165 should not be produced for these LSR pools. Consequently, CPExpert compares the total “hits” for each pool against the LSRINACT guidance variable. The LSRINACT guidance variable specifies the threshold below which CPExpert should ignore the “look-aside hits” because the pool is seldom used.

The default specification for this guidance variable is **%LET LSRINACT = 1000**, meaning that CPExpert will ignore the percent “look-aside hits” for any LSR pool that had less than 1000 total hits. You can provide different guidance to CPExpert by changing the LSRINACT variable.

Chapter 2.23: % I/O requests for LSR files - LSRIOREQ variable

VSAM files can be assigned as nonshared resources (NSR) files or assigned to a local shared resources (LSR) pool. The major difference between the two methods is how the VSAM strings and VSAM buffers are allocated and used.

- When CICS VSAM files are assigned as NSR files, there is no sharing of VSAM strings or VSAM buffers among files.

- When CICS VSAM files are assigned to LSR pools, the files share common strings and buffers assigned to the LSR pool.

There are some advantages to assigning files as NSR files, and there are major advantages to assigning files to LSR pools. The description of Rule CIC167 describes the advantages of each technique.

CPExpert analyzes the number of I/O requests for LSR files and computes this as a percent of all I/O requests for all VSAM files. CPExpert produces Rule CIC167 if this percent is less than the LSRIOREQ guidance variable.

The default of the LSRIOREQ guidance variable is 75, indicating that 75% of the I/O requests should be satisfied from files assigned to LSR pools. This default is set such that Rule CIC167 probably will initially be produced for many CICS regions. The point of producing Rule CIC167 is to alert you to the significant performance advantages of using LSR pools. If you are not inclined to assign more VSAM files to LSR pools, change the guidance variable and prevent the spurious production of Rule CIC167. (For example, you can "turn off" the rule completely by specifying **%LET LSRIOREQ = 0**.)

Chapter 2.24: Inactive pool - LSRUSE variable

VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times. Some files will have requirements for buffers at one time, while at another time they will not be accessed and will not require buffers. The demand for buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the pools. If VSAM files do not use the LSR pools that are created, the storage dedicated to a particular pool might be better used elsewhere in CICS.

CPExpert calculates the total I/O requests made to each LSR pool that was defined. The resulting value is compared with the LSRUSE guidance variable. CPExpert produces Rule CIC175 if the total number of I/O requests to the LSR pool is lower than the LSRUSE guidance variable. The default specification for the guidance variables is **%LET LSRUSE = 100**, meaning that CPExpert will produce Rule CIC175[3] when less than 100 I/O requests are made to the LSR pool. You can provide different guidance to CPExpert by changing the LSRUSE variable.

Chapter 2.25: Inactive subpool - LSRUSEn variables

LSR pools are generated using the DEFINE LSRPOOL command of Resource Definition Online, or using the DFHFCT TYPE=SHRCTL macro. The BUFFERS operand of this macro is used to define the number and size of the buffers assigned to the pool. Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the CI for data and index records of files assigned to the LSR pool. The pools with different buffer sizes are referred to as LSR subpools. There can be (and should be) different LSR subpools for data and index CI sizes.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the subpools. If VSAM files do not use the LSR subpools that are created, the storage dedicated to buffers for a particular subpool might be better used elsewhere in CICS.

The CICS statistics provide information about the number of times an I/O request was made to a LSR pool, and identify the subpool to which the request was made. CPExpert calculates the total I/O requests made to each LSR subpool that was defined. The

[3]CPExpert will not produce Rule CIC175 unless the condition exists for more than 50% of the CICS statistics intervals being analyzed.

resulting value is compared with the LSRUSEn guidance variable (where “n” is the number of the LSR pool, ranging from 1 to 8). CPExpert produces Rule CIC176 if the total number of I/O requests to the LSR subpool is lower than the LSRUSEn guidance variable.

The default value for each LSRUSEn guidance variable is based on the global LSRUSE guidance variable. The default values for the LSRUSE guidance variable is 100, meaning that CPExpert will produce Rule CIC176 when less than 100 I/O requests are made to the LSR subpool. You can use the default value for the LSRUSE guidance variable, specify a different value for this global variable (and have the global value applied to each LSR pool), or specify a different value for one or more LSR pools.

Chapter 2.26: Non-user buffer writes - NONUSRBF variable

User initiated buffer writes are the writing of buffers due to a direct request from the user program, such as EXEC CICS WRITE of a single record. This record must be written to the file immediately.

Non-user initiated buffer writes are usually encountered only when deferred requests are involved. For example, a MASSINSERT operation could cause non-user initiated buffer writes. Non-user initiated buffer writes normally mean that the number of buffers assigned to the LSR subpool should be increased.

CPExpert produces Rule CIC169 if the CICS interval statistics revealed that the number of non-user initiated buffer writes for any LSR subpool was greater than the **NONUSRBF** guidance variable. The default specification for this variable is **%LET NONUSRBF = 0**, indicating that non-user initiated buffer writes should not be encountered. You can provide different guidance to CPExpert by changing the NONUSRBF variable if you feel that Rule CIC169 is produced prematurely.

Chapter 2.27: Percent CICS-DB2 TCBs used - PCTD2TCB variable

The TCBLIMIT parameter specifies the maximum number of Task Control Blocks (TCBs) that can be used for a CICS-DB2 connection.

CPExpert produces Rule CIC275 when the percent peak TCBs in use is more than the value specified by the **PCTD2TCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2TCB** is 80 indicating that CPExpert should produce Rule CIC275 whenever the peak TCBs in use was more than 80% of the TCB limit specified for the CICS-DB2 Connection. You can provide different guidance to CPExpert by changing the PCTD2TCB variable if you feel that Rule CIC275 is produced prematurely.

Chapter 2.28: Percent CICS-DB2 threads used - PCTD2THR variable

The THREADLIMIT parameter specifies the maximum number of pool threads that can be used for a CICS-DB2 connection.

CPExpert produces Rule CIC277 when the percent pool threads in use is more than the value specified by the **PCTD2THR** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2THR** is 80 indicating that CPExpert should produce Rule CIC277 whenever the pool threads in use was more than 80% of the pool thread limit specified for the CICS-DB2 Connection. You can provide different guidance to CPExpert by changing the PCTD2THR variable if you feel that Rule CIC277 is produced prematurely.

Chapter 2.29: Percent EDSA used at peak usage - PCTEDSA variable

The EDSALIM keyword in the SIT specifies, in 1 megabyte increments, the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside above the 16MB boundary. From the storage value that is specified on the EDSALIM parameter, CICS allocates the extended user DSA (EUDSA), the extended read-only DSA (ERDSA), the extended shared DSA (ESDSA), and the extended CICS DSA (ECDSA).

When the amount of storage allocated by CICS reaches the limit specified by the EDSALIM keyword, the result can be increased program compression or, more seriously, SOS (short on storage) conditions, or even storage deadlock ABENDS when program compression is not sufficient.

The EDSALIM value (from the SIT or as modified with operator overrides) is contained in variable SMSEDSAL in the MXG file CICSMDSA. The CICS interval statistics contain information about the amount of EDSA allocated.

CPExpert compares the peak EDSA allocated against a percent of the value specified for the EDSALIM keyword (contained in variable SMSEDSAL). CPExpert produces Rule CIC113 when the peak EDSA allocated storage is greater than this percent. The percent used by CPExpert is specified by the **PCTEDSA** guidance variable in USOURCE(CICGUIDE).

The default value for the PCTEDSA guidance variable is 75%, indicating that Rule CIC113 should be produced when the peak EDSA allocated storage is above 75% of the value of the EDSALIM keyword in the SIT. You can provide different guidance to CPExpert by changing the PCTEDSA guidance variable if you feel that Rule CIC113 is produced prematurely.

Chapter 2.30: Percent EDSA used at peak usage - PCTEDSA variable

A CICS application accesses VSAM data sets using CICS file control commands. The file control commands can optionally specify that an update is to occur with the record being read (keyword UPDATE). The UPDATE option guarantees read integrity for the record. One result of this UPDATE option is that the record is locked (and, depending on the type of file, the control interval is locked). Another result of this UPDATE option is that, if the VSAM data set is assigned to a CICS-maintained data table, the VSAM source data set must be referenced by VSAM before the record is referenced in the data table reference.

Both results from using the UPDATE option cause overhead and potentially degrade performance. Consequently, the UPDATE option should be used only if the record is actually updated or it is deleted.

File control statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed a VSAM data set using the UPDATE option but did not subsequently update or delete the records.
CPExpert produces Rule CIC177 when the percent unnecessary UPDATE option is more than the value specified by the **PCTFCUPD** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTFCUPD** is 25 indicating that CPExpert should produce Rule CIC177 whenever more than 25% of the VSAM file accesses with the UPDATE option did not result in a corresponding change to the VSAM source data set.

CPExpert normally suppresses this finding if less than 500 GET UPDATE file control commands were issued against the file. You can specify a different threshold for suppressing Rule CIC177 by altering the FCGETUPD guidance variable in USOURCE(CICGUIDE).

Chapter 2.31: Active Tasks as Percent of MXT (PCTMXTHI)

The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Beginning with CICS Release 1.7, CICS will not create a Task Control Area (TCA) for a new task when the number of tasks concurrently in the region reaches the MXT value.

Specifying a “correct” value for the MXT keyword is a balance between (1) specifying a value that is too high and (2) specifying a value that is too low.

- If the MXT value is too high, storage is wasted and (with Goal Mode) unnecessary overhead is generated. The logic associated with Rule CIC104 deals with situation in which the MXT value is too high.

- If the MXT value is too low, CICS will fail to attach tasks when the number of tasks reaches the MXT value. Rule CIC101 provides an indication of the number of times that the number of active tasks reached the MXT value.

  This indication that MXT was reached is acceptable for many CICS regions. However, reaching MXT might be unacceptable for some critical CICS regions. Installation personnel need to be aware of the possibility of a too-low MXT value for critical CICS regions. The logic associated with Rule CIC108 can be used to provide an alert that MXT value might be too low.

CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is greater than the **PCTMXTHI** guidance variable, CPExpert concludes that the MXT value may be too low. Rule CIC108 is produced to report this conclusion.

The value of the PCTMXTHI guidance variable can be specified either as “nn%” or as “.nn” to indicate a percentage. The default value for the PCTMXTHI guidance variable is 100%, which effectively “turns off” the logic in Rule CIC108 (the percent cannot be higher than 100%). You can specify an appropriate percentage if you have critical CICS regions and wish to be alerted that the number of active tasks may be in danger of reaching the MXT specification. CPExpert will produce Rule CIC108 when the number of active tasks reaches the specified percent of MXT.

Chapter 2.32: Active Tasks as Percent of MXT (PCTMXTLO)

The value of the MXT keyword in the SIT controls the amount of space allocated by CICS for kernel stacks, and concurrently sets a limit on the maximum number of active tasks. Too high a value wastes storage and can cause CICS to become short-on-storage or experience other stress conditions. When executing in Goal Mode, a large MXT value can significantly increase overhead. CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is less than the **PCTMXTLO** guidance variable, CPExpert concludes that the MXT value may be too high. Rule CIC104 is produced to report this conclusion.

If the condition exists for **any** CICS statistics interval in the data being analyzed for a region, CPExpert lists the percent of active tasks as a function of all intervals, and places ‘***’ beside those intervals in which the conclusion was reached. This listing is done to give a sense of perspective of (1) how often and (2) when the condition was detected.

Initially, CPExpert arbitrarily produced Rule CIC104 if the number of active tasks was less than 75% of the MXT value. The **PCTMXTLO** value was introduced to allow users to specify guidance to CPExpert. The value of the PCTMXTLO guidance variable can be specified either as “nn%” or as “.nn” to indicate a percentage.

Chapter 2.33: Region is approaching maximum capacity (PCTQRTCB)

CICS always has two or three TCBs for normal processing, depending on the release of CICS:

- The quasi-reentrant (**QR** mode) TCB executes the quasi-reentrant application code and most CICS code. This TCB is available in all releases of CICS.
- The resource-owning (**RO** mode) TCB is used for opening and closing data sets and for program loading. This TCB is available in all releases of CICS.
- The file-owning (**FO** mode) TCB is used for opening and closing data sets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 1.

CICS optionally has other TCBs, depending on the release of CICS.

If the Monitoring Class option is ON (MNPER="ON" in the System Initialization Table), the dispatcher domain maintains an "accumulated CPU TCB time" in the DSGACT variable.

Additionally, the dispatcher maintains "accumulated time dispatched" and "accumulated time in MVS wait" variables (DSGTDT and DSGTWT, respectively). The sum of the *accumulated time dispatched* and *accumulated time in MVS wait* is approximately the elapsed time that CICS was operational.

Dividing the *accumulated CPU TCB time* by the sum of the *accumulated time dispatched* and *accumulated time in MVS wait* yields an approximation of the **percent CPU busy** of the CICS region.

The DSGACT value does not include uncaptured CPU time, so the value normally will be less than the amount of CPU time actually used by the CICS region. IBM's *CICS Performance Guides* state that even with a totally busy CICS region, the calculated percent CPU busy of the CICS region would not normally be 100%. Consequently, IBM suggests that a region should be considered approaching maximum capacity if the calculated percent CPU busy exceeds 70%.

CPExpert produces Rule CIC109 when the Percent CPU Busy is greater than the value specified by the **PCTQRTCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTQRTCB** guidance variable is 60% indicating that CPExpert should produce Rule CIC109 whenever more than the Percent CPU Busy was more than 60% for the CICS region. The default guidance value is less than the 70% suggested by IBM to give an "early warning" of a capacity restraint. You can provide different guidance to CPExpert by changing the PCTQRTCB variable if you feel that Rule CIC109 is produced prematurely.

Chapter 2.34: Waits for IMS PSB pool space - PSBWAIT variable |

The PSBPL operand in the SIT specifies the number of blocks in the CICS-DL/I program specification block (PSB) pool. The PSB pool is used only if the CICS-DL/I interface is used. CICS uses the value specified in the PSBPL operand to limit the total virtual storage allocated at any one time to the PSB pool. CICS does not reserve the amount of storage specified, but allocates and deallocates the storage as required.

If there is insufficient space in the PSB pool to handle PSB pool requests, an IMS routine is called to free the least-used buffers in the pool. In this case, the oldest PSB is deleted and the new PSB is loaded. The deletion (and particularly the loading) of PSBs slows the processing of tasks.

CPExpert produces Rule CIC194 if the number of waits for PSB pool space was greater than the **PSBWAIT** guidance variable, and if storage was not a constraint. The default specification for this guidance variable is **%LET PSBWAIT = 0**, indicating that Rule CIC194 will be produced if there were any waits for PSB pool space and storage was not constrained. You can provide different guidance to CPExpert by changing the PSBWAIT variable if you feel that Rule CIC194 is produced prematurely.

Chapter 2.35: Persistent Verification Timeouts - PVCOUNT variable

With CICS/ESA Release 3.2.1, "persistent verification" is the term used to describe signing on to a remote system and having that sign on remain valid (or "persist") over multiple conversations until it is no longer needed.

The PVDELAY parameter in the SIT is the "persistent verification" specification. The parameter specifies how long entries can remain signed on to the remote system when you are running remote transactions over ISC and IRC links.

Beginning with CICS/ESA Release 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the PVDELAY parameter (A21LUITM), provide the average time that has elapsed between each reuse of entries in the "signed on from" list (A21LUIAV), and provide the number of Persistent Verification (PV) timeouts (A21LUITI).

CPExpert produces Rule CIC255 if A21LUIAV is greater than A21LUITM, **and** A21LUITI is greater than the **PVCOUNT** guidance variable. The default specification for this variable is **%LET PVCOUNT = 0**, indicating that no persistent verification timeouts are acceptable. This low value is intended to alert you to the analysis process, and you probably will wish to change the value to a large number after your have reviewed the statistics. You can provide different guidance to CPExpert by changing the PVCOUNT variable if you feel that Rule CIC255 is produced prematurely.

Chapter 2.36: VTAM reached MAX RAPOOL - RAPOOL variable

Task input from a terminal is received by the VTAM access method and is passed to CICS, if CICS has a receive-any request outstanding. CICS receives data from VTAM in the receive-any input area (RAIA). The RAMAX operand in the SIT specifies the size in bytes of the I/O area that is allocated for each VTAM receive-any operation.

The RAPOOL operand in the System Initialization Table (SIT) specifies the number of concurrent receive-any requests that CICS is to process from VTAM. If RAPOOL is set too low, not all terminal messages may be processed during one dispatch of the terminal control program. This will cause transactions to be delayed and performance to suffer. If RAPOOL is set too high (and RAMAX is a large value), then storage is wasted.

CPExpert evaluates the "number of times reached maximum" in the VTAM statistics portion of the CICS requested or interval statistics. CPExpert produces Rule CIC150 if this value exceeds the **RAPOOL** guidance variable in USOURCE.CICGUIDE, and if storage was not constrained. The default specification for this guidance variable is **%LET RAPOOL = 0**. You can provide different guidance to CPExpert by changing the RAPOOL variable if you feel that Rule CIC150 is produced prematurely.

Chapter 2.37: SNT Timeouts - SNTCOUNT variable

The ISRDELAY parameter in the SIT is the "intersystem refresh delay" specification. The parameter specifies how long entries can remain signed on to the remote system when you are running remote transactions over ISC and IRC links.

Beginning with CICS/ESA Release 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the ISRDELAY parameter (A21SNTTM), provide the average time that has elapsed between each reuse of userids (A21SNTAV), and provide the number of Signon Table (SNT) timeouts (A21SNTTI).

CPExpert produces Rule CIC254 if A21SNTAV is greater than A21SNTTM, **and** A21SNTTI is greater than the **SNTCOUNT** guidance variable. The default specification for this variable is **%LET SNTCOUNT = 0**, indicating that no SNT timeouts are acceptable. This low value is intended to alert you to the analysis process, and you probably will wish to change the value to a large number after your have reviewed the statistics. You can provide different guidance to CPExpert by changing the SNTCOUNT variable if you feel that Rule CIC254 is produced prematurely.

Chapter 2.38: Number of storage dumps - STORDUMP variable

Storage dumps are produced for a variety of reasons (e.g., program checks and storage violations). The system is severely loaded while writing a dump to the CICS dump data set. In the case of formatted dumps, nothing else is processed by CICS until the dump is written to the CICS dump data set. Depending upon the size of the dump, the dump could take over one second. During this time, all other activities are delayed.

CPExpert produces Rule CIC130 if the number of storage dumps is greater than the **STORDUMP** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET STORDUMP = 1**. You can provide different guidance to CPExpert by changing the STORDUMP variable if you feel that Rule CIC130 is produced prematurely.

Chapter 2.39: Waiting for VSAM Strings - STRWAIT variable

A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built. If multiple strings are available, multiple VSAM requests can be executed concurrently. If multiple VSAM requests are executed concurrently, the buffers will be transferred to DASD quicker, since several I/O requests can be outstanding at one time.

The CICS interval statistics provide information about the number of times tasks were required to wait for strings. This information is provided for each VSAM file (summary information is provided for LSR pools).

CPExpert divides the wait-on-string values in the CICS interval statistics by the total file access operations. CPExpert produces Rule CIC160 if the resulting percentage is greater than the **STRWAIT** guidance variable. The default specification for this guidance variable is **%LET STRWAIT = 5**, indicating that Rule CIC160 should be produced if more than 5% of the VSAM file I/O operations wait for VSAM strings.

The CICS Performance Guide indicates that it may be acceptable to have 5% of the file accesses for NSR files waiting on strings, but no guidance is provided for LSR files. You can provide different guidance to CPExpert by changing the STRWAIT variable.

Chapter 2.40: Number of transaction errors - TRANSERR variable

Transaction errors are errors in which the transaction associated with a particular terminal could not be started. This means that:

- A transaction identifier had not been defined in the CICS System Definition (CSD) data set or in the Program Control Table (DFHPCT).

- The operator did not have the proper security access to enter the transaction.

- The transaction had been disabled.

CPExpert produces Rule CIC140 when the number of transaction errors for any particular terminal exceeds the **TRANSERR** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET TRANSERR = 0**. You can provide different guidance to CPExpert by changing the TRANSERR variable if you feel that Rule CIC140 is produced prematurely.

Chapter 2.41: Shipped terminal deletes - TRMSKDEL variable

In a transaction routing environment, terminal definitions can be "shipped" from a terminal-owning region (TOR) to an application-owning region (AOR). The terminal definition becomes redundant in the AOR under the certain conditions. Shipped terminal definitions which have become redundant may need to be deleted. The CICS-supplied transaction CRMF periodically scans the shipped terminal definitions in the AOR and flags those which it has determined to be redundant. The system initialization parameter DSHIPINT controls the frequency at which shipped terminal definitions are tested for redundancy. If the default value of 12 hours (or a similarly high value) is specified for the DSHIPINT parameter, the CRMF transaction might identify a considerable number of redundant terminal definitions during a single expiration of the DSHIPINT value. Consequently, a relatively large “burst” of CPU would be required for the CRMD transaction processing.

CPExpert compares the A04SKDEL variable in CICSAUTO (the number of shipped terminals deleted) with the **TRMSKDEL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC156 when the number of shipped terminals deleted exceeds the value specified by the **TRMSKDEL** guidance variable.

The default value for the **TRMSKDEL** guidance value is 50, indicating that Rule CIC156 will be produced when more than 50 shipped terminal definitions have been deleted during any CICS interval statistics period. You can change this guidance if you fell that Rule CIC156 is produced prematurely.

Chapter 2.42: % TS I/O requests buffer wait - TSIOWAIT variable

VSAM multiple buffers allow multiple VSAM control intervals to be available in storage at the same time. When multiple buffers are available for temporary storage, the CICS temporary storage programs can use different buffers to respond to several concurrent requests for temporary storage. Using multiple VSAM buffers also increases the probability that the control interval required by any particular request will already be available in a buffer. If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and tasks are queued serially by temporary storage queue name.

It is generally desirable to minimize the number of times tasks must wait for access to a VSAM buffer. When the tasks wait, they continue to hold system resources (e.g., real storage), and may cause additional performance problems because the resources are not available to service other tasks.

CPExpert computes the percent of temporary storage I/O requests that waited for VSAM buffers by dividing the number of waits by the number of temporary storage I/O requests. CPExpert produces Rule CIC186 if this percent exceeds the **TSIOWAIT** guidance variable in USOURCE.CICGUIDE. The default specification for this guidance variable is **%LET TSIOWAIT = 5**, based upon guidelines in IBM's CICS Performance Guide. You can provide different guidance to CPExpert by changing the TSIOWAIT variable.

Chapter 3: Specifying guidance for individual CICS regions

This chapter applies only if you analyze multiple CICS regions during a single execution of the CICS Component of CPExpert and wish to provide unique guidance to certain CICS regions.

Some users of the CICS Component use CPExpert to analyze dozens or hundreds of CICS regions. These regions provide service to different users, the regions often have different applications, and the regions may even be a mixture of test regions and production regions. One set of analysis guidance variables do not always apply to all regions being analyzed. Consequently, CPExpert provides the capability to override the basic guidance on a region-by-region basis. This chapter describes how to implement this override guidance.

Unique guidance for particular CICS regions can be accomplished using the **GUIDE** guidance variable and the **applid_x = GUIDE** guidance variables, placed in the *ANALYSIS CONTROL VARIABLES* section of USOURCE(CICGUIDE), and placing the unique guidance into a USOURCE(applid_x) PDS member. Exhibit 3-2 describes the *ANALYSIS CONTROL VARIABLES* section.

Chapter 3.1: Specify GUIDE guidance variable

The **GUIDE** guidance variable specifies whether override guidance exists for one or more regions.

CPExpert normally uses the guidance in USOURCE(CICGUIDE) to control its analysis of CICS regions. You can override this process by specifying **%LET GUIDE = Y;** in the *ANALYSIS CONTROL VARIABLES* section of USOURCE(CICGUIDE). When you make this specification, CPExpert generates code to acquire additional guidance for particular CICS regions, as controlled by the **applid_x = GUIDE** guidance variable(s).

The purpose of having a “general” controlling parameter is to provide a “switch” that allows you to normally use override guidance, but you can easily turn off the overrides and analyze all regions using the basic guidance.

Chapter 3.2: Identify specific CICS regions having unique guidance

Each CICS region for which you wish to provide specific guidance must be identified to CPExpert. This identification is done by specifying **%LET applid_x = GUIDE;** in USOURCE(CICGUIDE). For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would specify **%LET CICSPVA7=GUIDE;** in USOURCE(CICGUIDE). This specification tells CPExpert that unique guidance for APPLID CICSPVA7 is contained in USOURCE(CICSPVA7).

Chapter 3.3: Place unique guidance in USOURCE(applid) member

Create a unique analysis guidance member in the USOURCE partitioned data set. For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would create a USOURCE(CICSPVA7) member.

You can specify any unique guidance for the particular APPLID by entering the appropriate SAS %LET statements into the USOURCE(applid) member. For example, if you wish to specify unique guidance for APPLID CICSPVA7, you would specify appropriate SAS %LET statements in USOURCE(CICSPVA7).

Important: If you should simply copy the USOURCE(CICGUIDE) member to USOURCE(applid) member, you must delete from the new module all information unrelated to the unique guidance. **Do not retain data selection and reporting variables**! Unpredictable analysis could result if you retain the data selection and reporting variables.

Chapter 3.4: Restrictions

The following restrictions apply to specifying unique guidance for CICS regions when processing multiple CICS regions during a single execution of the CICS Component:

- You can specify unique guidance for up to 99 CICS regions.
- The override guidance specific to a particular CICS region applies only to that region. CPExpert reverts to the original guidance contained in USOURCE(CICGUIDE) for subsequent CICS regions (unless those CICS regions have unique guidance).
- You will get a SAS error if you specify %LET applid_x=GUIDE; and do not have the corresponding member as USOURCE(applid_x).
- You can “comment out” any %LET applid_x=GUIDE; statement simply by replacing the “%” with “*” (that is, replace the percent with an asterisk)..

Chapter 4: Analyzing CICS report classes

Some users of the CICS Component want CPExpert to provide a "warning" about certain system-level resource use or performance characteristics related to specific CICS workload. To meet this requirement, the CICS Component has been enhanced to provide the ability to associate specific CICS workload with system-level information contained in RMF Type72 records.

The initial requirement was for CPExpert to provide a warning when certain CICS workload used more than a specified percent of CPU[4]. For this requirement, logic was implemented to associate specific CICS workload with the MXG TYPE72GO information, and guidance was implemented using the CPUWARN guidance variable. While other features may be implemented, the "CPU warning" construct will be used to discuss and illustrate both the concept and the implementation of the enhancement.

The "CPU warning" approach requires that a Report Class describing the CICS workload of interest be defined to the Workload Manger (WLM). This Report Class is defined using the standard WLM workload classification panel.

During CICS operation, the System Resources Manager (SRM) acquires information related to the defined Report Class, and RMF writes the Report Class information[5] to SMF as part of the SMF Type 72 records. MXG processes the SMF Type 72 records and places the SMF information into various files in the MXG performance data base. CPExpert processes the MXG TYPE72GO file to extract information about the defined Report Class. CPExpert produces a "CPU warning" when the defined Report Class uses more than "nn%" CPU during any RMF reporting interval.

Chapter 4.1: Implementing analysis of CICS report classes

A user of the CICS Component implements the Report Class analysis feature by the following steps:

---

[4]This feature is similar to the analysis described in Rule CIC109; however, Rule CIC109 relates to the quasi-reentrant (QR) TCB time, which is considered by IBM to be not as representative of the CPU time required by the region as would be data produced by RMF. For a warning regarding CPU time used by a region, either the "QR TCB approach" described in Rule CIC109 or the "report class approach" described by this discussion probably would yield similar results.

[5]The data available for report classes include:

- Number of transactions completed
- Average response times
- Resource usage data
- State samples.

Additional data was added to report class information, beginning with z/OS Version 1, Release 2:

- Response time distribution buckets.
- Work manager delay data.

- **Step 1. Define Report Class**. The "Defining Report Classes" section of IBM's *MVS Planning: Workload Management* document describes how to define report classes to the WLM. In brief, the WLM's classification rules can assign incoming work to a report class. Report classes can be used to report on a subset of transactions running in a single service class, or can can be used to report on a combination of transactions running in different service classes within one report class.

- **Step 2: Specify that report classes are to be analyzed by the CICS Component**. The **RPRTCLASS** guidance variable is used to tell CPExpert that report classes are to be analyzed. If you wish the CICS Component to analyze report classes, specify **%LET RPTCLASS=Y;** in USOURCE(CICGUIDE).

- **Step 3: Identify the report classes to be analyzed**. There can be many report classes defined using the WLM Report Class Definition panel. You must tell CPExpert which specific report classes are to be analyzed by the CICS Component. This is done by specifying **%LET RPRTn=report_class_name;** in USOURCE(CICGUIDE). For example, to tell CPExpert that the CICSAOR3 report class should be analyzed, specify **%LET RPRT1=CICSAOR3;** in USOURCE(CICGUIDE). You can specify up to 99 report classes[6] to be analyzed by the CICS Component.

Once the above steps have been completed, the CICS Component will process MXG TYPE72GO, selecting records that meet the overall data selection criteria (SYSTEM, start/end dates and times, etc.), and that are for the defined report classes. Subsequent analysis will apply the appropriate evaluation logic (e.g., when the CICS workload identified by the report classes used more than the percent of CPU specified by the CPUWARN guidance variable). The CIC700(series) rules will be produced when the resource use exceeds the guidance.

Chapter 4.2: Specifying guidance for individual CICS report classes

The CICS Component can be used to analyze resource use or performance characteristics of multiple CICS report classes during a single execution of the CICS Component. One set of analysis guidance variables does not always apply to all CICS report classes being analyzed. Consequently, CPExpert provides the capability to override the basic guidance on a report-class-by-report-class basis. This chapter describes how to implement this override guidance.

Chapter 4.2.1: Specify RPTGUIDE guidance variable

The **RPTGUIDE** guidance variable specifies whether override guidance exists for one or more report classes.

---

[6]Please notify Don_Deese@cpexpert.com if you require more than 99 report classes.

CPExpert normally uses the guidance in USOURCE(CICGUIDE) to control its analysis of CICS report classes. You can override this process by specifying **%LET RPTGUIDE = Y;** in the *ANALYSIS* *CONTROL* *VARIABLES* section of USOURCE(CICGUIDE). When you make this specification, CPExpert generates code to acquire additional guidance for particular CICS regions, as controlled by the **report_class_x = GUIDE** guidance variable(s).

The purpose of having a "general" controlling parameter is to provide a "switch" that allows you to normally use override guidance, but you can easily turn off the overrides and analyze all regions using the basic guidance.

Chapter 4.2.2: Identify specific report classes having unique guidance

Each CICS report class for which you wish to provide specific guidance must be identified to CPExpert. This identification is done by specifying **%LET report_class_x = GUIDE;** in USOURCE(CICGUIDE). For example, if you wish to specify unique guidance for CICS Report Class CICSAOR3, you would specify **%LET CICSAOR3=GUIDE;** in USOURCE(CICGUIDE). This specification tells CPExpert that unique guidance for Report Class CICSAOR3 is contained in USOURCE(CICSAOR3).

Chapter 4.2.3: Place guidance in USOURCE(report_class) member

Create a unique analysis guidance member in the USOURCE partitioned data set. For example, if you wish to specify unique guidance for Report Class CICSAOR3, you would create a USOURCE(CICSAOR3) member.

You can specify any unique guidance for the particular CICS report class by entering the appropriate SAS %LET statements into the USOURCE(report_class) member. For example, if you wish to specify unique guidance for Report Class CICSAOR3, you would specify appropriate SAS %LET statements in USOURCE(CICSAOR3).

Important: If you should simply copy the USOURCE(CICGUIDE) member to USOURCE(applid) member, you must delete from the new module all information unrelated to the unique guidance. **Do not retain data selection and reporting variables**! Unpredictable analysis could result if you retain the data selection and reporting variables.

Chapter 4.2.4: Restrictions

The following restrictions apply to specifying unique guidance for CICS report classes when processing multiple CICS report classes during a single execution of the CICS Component:

- You can specify unique guidance for up to 99 CICS report classes.

- The override guidance specific to a particular CICS report class applies only to that report class. CPExpert reverts to the original guidance contained in USOURCE(CICGUIDE) for subsequent CICS report classes (unless those CICS report classes have unique guidance).

- You will get a SAS error if you specify %LET report_class_x=GUIDE; and do not have the corresponding member as USOURCE(report_class_x).

- You can "comment out" any %LET report_class_x=GUIDE; statement simply by replacing the "%" with "*" (that is, replace the percent with an asterisk).

Chapter 5: 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 Release 1.3 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

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 off loaded 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[7] being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[8]. 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.

[7] 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.

[8] 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 3-3 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes system logger performance. This chapter describes these variables, how the variables are used, and how the variables are altered.

```
*************************************************************************    ;
*        CICS 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 PCTINST  =0        ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USE;
%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  ;
%LET STRFULL  =0        ; * ACCEPTABLE LOG STREAM STRUCTURE FULL      ;

%LET LOGGUIDE = N       ; * GUIDANCE PROVIDED FOR INDIVIDUAL LOG STRMS ;
/* SPECIFY GUIDANCE FOR LOG STREAMS: EXAMPLE
LOGNAME = log.stream.name1
%LET PCTINST  =0        ; * PERCENT INTERIM STORAGE NOT EFFECTIVELY USE;
%LET PCTLOCST =0        ; * PERCENT LOCAL STORAGE NOT EFFECTIVELY USED ;
%LET STFULL90 =0        ; * ACCEPTABLE STRUCTURE 90% FULL             ;
%LET STDSHIGH =0        ; * STAGING DATA SET HIGH THRESHOLD HIT       ;
*/
```

DEFAULT VALUES FOR SYSTEM LOGGER ANALYSIS

EXHIBIT 3-3

Chapter 5.1: SMF Type 88 records available - SMFTYP88 variable

The SMFTYP88 guidance variable is used to tell CPExpert whether you wish to analyze system logger performance based on SMF Type 88 records. If the SMFTYP88 variable is **N**, CPExpert will **not** analyze system logger performance based on SMF Type 88 records. If the SMFTYP88 variable is **Y**, CPExpert **will** analyze system logger performance based on SMF Type 88 records.

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.

Chapter 5.2: Log stream staging data set full - LGDSFULL variable

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(CICGUIDE). CPExpert produces Rule CIC302 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 CIC302 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 5.3: Log stream DASD-shift conditions - LGSHIFTS variable

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 CIC307 when the number of DASD shifts that occur during any hour exceeds the **LGSHIFTS** guidance variable in USOURCE(CICGUIDE). The default value for the **LGSHIFTS** is one, indicating that CPExpert should produce Rule CIC307 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 5.2: Percent interim storage offloaded - PCTINTST variable

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[9], 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.

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

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 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(CICGUIDE). CPExpert produces Rule CIC304 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 CIC304 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 5.3: Percent use of staging data sets - PCTLOCT variable |

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[10], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, |

[10] Interim storage is sometimes referred to as “primary” 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 residing in coupling facility structures is analyzed by Rule CIC304 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 CIC305 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[11]. 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.

[11] 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 CIC305.

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

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC305 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 CIC305 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 5.4: Staging data set threshold - STDSHIGH variable

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[12].

CPExpert examines the SMF88STN variable (the structure name) in the MXG TYPE88 data set to select records that apply only to coupling facility structures[13]. 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 CIC306 when the SMF88ETF value exceeds the **STDSHIGH** guidance variable in USOURCE(CICGUIDE).

The default value for the **STDSHIGH** is zero, indicating that CPExpert should produce Rule CIC306 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 5.5: Log stream structure 90% full - STFULL90 variable

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.

[12] 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 CIC302.

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

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%).

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[14]. 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 CIC303 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 5.6: Log stream Type-2 completions - STRC2 variable

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.

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

- 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 structure to DASD log data sets or if the system logger configuration is tuned incorrectly. The Type-3 completions are analyzed by Rule CIC309.

CPExpert examines the SMF88SC2 variable (Count of Type-2 completions) in the SMF Type 88 records. CPExpert produces Rule CIC308 when the SMF88SC2 value exceeds the **STRC2** guidance variable in USOURCE(CICGUIDE).

The default value for the **STRC2** is zero, indicating that CPExpert should produce Rule CIC308 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 5.7: Log stream Type-3 completions - STRC3 variable |

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 CIC308.

- 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 CIC309 when the SMF88SC3 value exceeds the **STRC3** guidance variable in USOURCE(CICGUIDE).

The default value for the **STRC3** is zero, indicating that CPExpert should produce Rule CIC309 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 5.8: Log stream CF structure full - STRFULL variable

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 Release 2.4 (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[15]. This condition is analyzed by Rule CIC301.

- 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

[15] 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!

has reached its capacity before off loading data to secondary storage. This condition is analyzed by Rule CIC302.

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 TYPE88 data set with the **STRFULL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC301 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 CIC301 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 5.9: 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 from the perspective of CICS. 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 the 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(CICGUIDE), identifying the log stream(s) or log stream structure(s) to which the guidance applies, and specifying guidance variables for the log stream(s) or log stream structure(s).

Please note that the guidance is specified for specific log streams or log stream structures. The guidance is NOT specified for specific CICS regions, nor may guidance for specific log streams or log stream structures be "overridden" on a region-by-region basis.

From a practical matter, this limitation should not pose a problem, since the log stream names normally correspond to specific CICS regions. Thus, specifying guidance for a particular log stream or log stream structure typically would cause the guidance to be associated with a CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific log streams. As illustrated in Exhibit 3-4, guidance for individual log streams 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(CICGUIDE) member.

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

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

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

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 CIC308).

```
%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          ;
*/
```

Chapter 6: Shared Temporary Storage Guidance Variables

The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct access storage device, or in a temporary *storage data sharing pool*. Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a temporary storage *data sharing server* that supports the named pool. In each MVS image in the sysplex, there is one data sharing server for each pool defined in a coupling facility which can be accessed from that MVS image. All TS pool access is performed by cross-memory calls to the data sharing server for the named pool.

The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITE TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

During server initialization, the server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. It also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. After sharing server initialization, AXM page allocation services are used to manage server region storage.

CICS provides statistics related to the shared temporary storage queue server:

- Shared temporary storage queue server statistics for the coupling *facility* are available in MXG file **CICXQ1**.
- Shared temporary storage queue server *buffer pool statistics* available in MXG file **CICXQ2**.
- Shared temporary storage queue server *storage statistics* available in MXG file **CICXQ3**.

The CICS Component analyzes data in CICXQ1, CICXQ2, and CICXQ3 to detect performance problems or potential performance problems with shared temporary storage. Exhibit 3-4 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes shared temporary storage. This chapter describes the guidance variables that can be used to alter CPExpert's analysis of shared temporary storage.

```
**************************************************************************    ;
*        CICS Component GUIDANCE VARIABLES                    ;
**************************************************************************    ;
      •
      •
      •
      •

* SHARED TS QUEUE SERVER GUIDANCE VARIABLES;
*;
* SERVER COUPLING FACILITY STATISTICS GUIDANCE;
%LET TSLSTFUL = 0       ; * TIMES LIST WAS FULL                         ;
%LET TSNOSPCE = 0       ; * TIMES LIST STRUCTURE WAS OUT OF SPACE     ;
%LET TSPCTELE = 70      ; * MAXIMUM PERCENT DATA ELEMENTS USED        ;
%LET TSPCTENT = 70      ; * MAXIMUM PERCENT LIST ENTRIES USED         ;
%LET TSPCTIDR = 0       ; * MINIMUM PERCENT STRUC INDEX DATA REREADS  ;
%LET TSPCTLDR = 0       ; * MINIMUM PERCENT STRUC LIST DATA REREADS   ;
%LET TSPCTNOE = 0.1     ; * PERCENT STRUCTURE ENTRY NOT FOUND         ;
%LET TSPCTTIM = 25      ; * PERCENT REQUESTS TIMEOUT AND RESTARTED    ;
%LET TSPCTVCF = 1       ; * MAXIMUM PERCENT VERSION CHECK FAILED      ;
* SERVER BUFFER POOL STATISTICS GUIDANCE;
%LET TSPCTFBP = 75      ; * PERCENT QUEUE INDEX BP BUFFERS USED       ;
%LET TSPCTLRU = 0.1     ; * PERCENT LRU ACTIVITY                      ;
%LET TSPCTUSE = 10      ; * PERCENT BUFFERS USED IN INDEX BUFFER POOL ;
%LET TSPCTWBL = 1       ; * PERCENT WAIT ON BUFFER LOCK               ;
%LET TSPCTWBP = 0.1     ; * PERCENT WAIT ON BUFFER POOL LOCK          ;
* SERVER STORAGE STATISTICS GUIDANCE;
%LET TSANYRQS = 0       ; * LOC=ANY STORAGE REQUEST FAILED AFTER RETRY;
%LET TSLOWRQS = 0       ; * LOC=LOW STORAGE REQUEST FAILED AFTER RETRY;
%LET TSPCTAMN = 1       ; * LOC=ANY PERCENT MINIMUM FREE STORAGE      ;
%LET TSPCTARC = 0.1     ; * LOC=ANY PERCENT REQUEST FAILED AND RETRIED;
%LET TSPCTLMN = 1       ; * LOC=LOW PERCENT MINIMUM FREE STORAGE      ;
%LET TSPCTLRC = 0.1     ; * LOC=LOW PERCENT REQUEST FAILED AND RETRIED;

**********************************************************************;
```

DEFAULT VALUES FOR SHARED TEMPORARY STORAGE ANALYSIS

EXHIBIT 3-4

Guidance variables for shared temporary storage queue server are described in three part: (1) server coupling facility statistics guidance variables, (2) server buffer pool statistics guidance variables, and (3) server storage statistics guidance variables.

Chapter 6.1: Server coupling facility statistics |

Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region. Shared temporary storage queue server *coupling facility statistics* are available in MXG file CICXQ1.

Data items in shared temporary storage (TS) are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. These queues are related to a shared TS pool using the SYSID keyword to specify the shared queue pool to which the request is directed. Each TS pool is defined, using MVS cross-system extended services (XES), as a keyed list structure in a coupling facility.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With shared TS queues, the list structure is the named pool, while the lists themselves are shared TS queues within the named pool. The shared TS pool server designates the maximum number of lists (or queues) the TS pool is to have, and allocates the list structure based on parameters that are provided to the TS pool server.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both. Data entries are composed of units of storage called *data elements*. Although a data entry can be composed of a number of data elements, list operations treat the data entry as a single entity; data elements cannot be read or written individually.

Chapter 6.1.1: Times list was full - TSLSTFUL variable |

Three parameters control how many shared temporary storage queues can be in a specific pool and the characteristics of the queues:

- The *MAXQUEUES* parameter specifies the maximum number of data lists to be reserved when the structure is allocated.

- The *SMALLQUEUEITEMS* specifies the maximum number of items that can be stored in the small queue format in the queue index entry data area.

- The *SMALLQUEUESIZE* parameter specifies the maximum data size for a small queue including the two-byte length prefix on each data item.

If the number of queues is reached (as set by the MAXQUEUES server initialization parameter described above), any further request to establish a queue will fail, and message **DFHXQ0443** (*CF structure strname request failed, all lists are in use*) will be issued. The failing request is given a NOSPACE indication if it originated from a CICS API request.

CPExpert uses data in CICXQ1 to determine whether a List Full condition occurred. CICXQ1 variable S1RSP6CT (List full: maximum list key reached) indicates that maximum queue size or maximum queues were reached, depending on the list.

CPExpert produces Rule CIC324 when the number of List Full conditions is greater than the value specified by the **TSLSTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSLSTFUL** is 0, indicating that CPExpert should produce Rule CIC324 whenever any List Full conditions occurred.

You can alter this analysis using the **TSLSTFUL** guidance variable. For example, if you wish Rule CIC324 produced only when more than one List Full condition occurred, specify:

```
%LET TSLSTFUL = 1 ; * TIMES LIST WAS FULL;
```

Chapter 6.1.2: List structure was out of space - TSNOSPCE variable

If a task tries to write to temporary storage and there is no space available, message **DFHXQ0442** (*CF structure strname request failed, structure is full*) will be issued. The failing request is given a NOSPACE indication if it originated from a CICS API request. CICS normally suspends the task (although the task can regain control in this situation by using either a HANDLE CONDITION NOSPACE command, or the RESP or NOHANDLE option on the WRITEQ TS command). If suspended, the task normally is not resumed until some other task frees the necessary space in main storage or the VSAM data set.

CPExpert uses data in CICXQ1 to determine whether a Structure Full condition occurred. CICXQ1 variable S1RSP7CT (List structure out of space) indicates that the list structure was full.

CPExpert produces Rule CIC325 when the number of Structure Full conditions is greater than the value specified by the **TSNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSNOSPCE** is 0, indicating that CPExpert should produce Rule CIC325 whenever any Structure Full conditions occurred.

You can alter this analysis using the **TSNOSPCE** guidance variable. For example, if you wish Rule CIC324 produced only when more than one Structure Full condition occurred, specify:

```
%LET TSNOSPCE = 1 ; * TIMES LIST STRUCTURE WAS OUT OF SPACE;
```

Chapter 6.1.3: Percent data elements in use - TSPCTELE variable

With shared temporary storage, the queue server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued. The default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CPExpert uses data in CICXQ1 to calculate the maximum percent of the data elements that had been used.

CPExpert produces Rule CIC321 when the maximum percent data elements used is more than the value specified by the **TSPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTELE** is 70 indicating that CPExpert should produce Rule CIC321 whenever more than 70% of the data elements had been used.

You can alter this analysis using the **TSPCTELE** guidance variable. For example, if you wish Rule CIC321 produced only when more than 60% of the data elements had been used, specify:

```
%LET TSPCTELE = 60 ; * MAXIMUM PERCENT DATA ELEMENTS USED;
```

Chapter 6.1.4: Percent list entries in use - TSPCTENT variable

With shared temporary storage, the queue server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CPExpert uses data in

CICXQ1 to calculate the maximum percent of the structure list entries that had been used.

CPExpert produces Rule CIC320 when the maximum percent structure list entries used is more than the value specified by the **TSPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTENT** is 70 indicating that CPExpert should produce Rule CIC320 whenever more than 70% of the list entries had been used.

You can alter this analysis using the **TSPCTENT** guidance variable. For example, if you wish Rule CIC320 produced only when more than 60% of the list entries had been used, specify:

```
%LET TSPCTENT = 60 ; * MAXIMUM PERCENT LIST ENTRIES USED;
```

Chapter 6.1.5: Percent repeated index data reads - TSPCTIDR variable

The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

CPExpert uses data in CICXQ1 to calculate the percent of *index data* reads which had to be repeated because the data was larger than the default data transfer size.

CPExpert produces Rule CIC327 when the percent of queue index reads had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTIDR** guidance variable. The default value for the **TSPCTIDR** is 0, indicating that CPExpert should produce Rule CIC327 when any queue index reads must be repeated because the data was larger than the default data transfer size.

You can alter this analysis using the **TSPCTIDR** guidance variable. For example, if you wish Rule CIC327 produced only when more than 10% of the queue index reads must be repeated, specify:

```
%LET TSPCTIDR = 10 ; * MINIMUM PERCENT STRUCTURE INDEX DATA REREADS;
```

Chapter 6.1.6: Percent repeated list data reads - TSPCTLDR variable

The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of list data reads which had to be repeated because the data was larger than the default data transfer size.

CPExpert produces Rule CIC326 when the percent of list data reads which had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTLDR** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLDR** is 0 indicating that CPExpert should produce Rule CIC326 when any list data reads must be repeated because the data was larger than the default data transfer size.

You can alter this analysis using the **TSPCTLDR** guidance variable. For example, if you wish Rule CIC326 produced only when more than 10% of the list data reads must be repeated, specify:

```
%LET TSPCTLDR = 10 ; * MINIMUM PERCENT STRUCTURE LIST DATA REREADS;
```

Chapter 6.1.7: Percent structure entry not found - TSPCTNOE variable

The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct access storage device, or in a temporary storage data sharing pool. Several commands can be used to reference, modify, or delete data in a temporary storage queue or to delete the entire queue.

When the commands are executed, the response from the command can be "normal" in that no further action is required, or the response can be "abnormal" in that either the task is terminated or action is required. Two abnormal conditions that apply to shared temporary storage are reported in the CICS statistics: *ITEMERR* and *QIDERR*.

- The ITEMERR abnormal condition can occur with the READQ TS command when (1) the item number specified is invalid (that is, the item number is outside the range of item numbers written to the queue), or (2) an attempt is made to read beyond the end of the queue using the NEXT (default) option.

- The QIDERR abnormal condition can occur with the READQ TS command, with the WRITEQ TS command, and with the DELETEQ TS command. The QIDERR abnormal condition occurs when the specified queue cannot be found.

CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a "specified entry (queue or item) was not found" condition.

CPExpert produces Rule CIC323 when the percent requests that encountered a "specified entry (queue or item) was not found" condition is greater than the value specified by the **TSPCTNOE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTNOE** is 0.1 indicating that CPExpert should produce Rule CIC323 whenever more than one tenth percent of the requests encountered a "specified entry (queue or item) was not found" condition.

You can alter this analysis using the **TSPCTNOE** guidance variable. For example, if you wish Rule CIC323 produced only when more than 10% of the requests encountered a "specified entry (queue or item) was not found" condition, specify:


```
%LET TSPCTNOE = 10 ; * PERCENT REQUESTS STRUCTURE ENTRY NOT FOUND;
```

Chapter 6.1.8: Percent requests timeout - TSPCTTIM variable

Some list structure commands can complete prematurely, because the request exceeded the coupling facility model-dependent time-out criteria. When list structure commands complete prematurely, the application typically restarts the command (using information has been returned in the answer area). Commands that complete prematurely cause unnecessary overhead in the coupling facility and in the application.

CPExpert uses data in CICXQ1 to calculate the percent of requests that timed out. CPExpert produces Rule CIC322 when the percent requests that timed out is greater than the value specified by the **TSPCTTIM** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTTIM** is 1 indicating that CPExpert should produce Rule CIC322 whenever more than one percent of the requests timed out because the request exceeded the coupling facility model-dependent time-out criteria.

You can alter this analysis using the **TSPCTTIM** guidance variable. For example, if you wish Rule CIC322 produced only when more than 10% of the requests timed out, specify:

```
%LET TSPCTTIM = 10 ; * PERCENT REQUESTS TIMEOUT AND RESTARTED;
```

Chapter 6.1.9: Percent version check failed - TSPCTVCF variable

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke implicit enqueuing. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

After issuing the READQ TS command, if the application wishes to modify the information and then issue a WRITEQ TS command, it is possible that the temporary storage queue record would have been updated by another task. In this case, the WRITEQ TS command would fail because of a version check.

CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a “version check failed for an entry being updated” condition.

CPExpert produces Rule CIC328 when the percent requests that encountered a “version check failed for an entry being updated” condition is greater than the value specified by the **TSPCTVCF** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTVCF** is 0.1 indicating that CPExpert should produce Rule CIC328 whenever more than one tenth percent of the requests encountered a “specified entry (queue or item) was not found” condition.

You can alter this analysis using the **TSPCTVCF** guidance variable. For example, if you wish Rule CIC323 produced only when more than 10% of the requests encountered a “version check failed for an entry being updated” condition, specify:

```
%LET TSPCTVCF = 10 ; * PERCENT VERSION CHECK FAILED CONDITION;
```

Chapter 6.2: Server buffer pool statistics

The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITEQ TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

When a request for the same queue arrives, the shared TS queue server determines whether the queue index information is in the buffer. If the information is in the buffer, a coupling facility access is avoided. When the request completes, the shared TS queue server places the information into a buffer, onto a least recently used (LRU) chain. If all other buffers are in use, a request for a new buffer will discard the contents of the least recently used buffer and reuse the storage as a free buffer.

Shared temporary storage queue server *buffer statistics* are available in MXG file CICXQ2.

Chapter 6.2.1: Percent index buffers in use - TSPCTFBP variable

If a large percent of buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner.

CPExpert uses data in CICXQ2 to calculate the percent of queue index buffer pool buffers that were used.

CPExpert produces Rule CIC330 when the percent queue index buffer pool buffers that were used is greater than the value specified by the **TSPCTFBP** guidance variable in

USOURCE(CICGUIDE). The default value for the **TSPCTFBP** is 75, indicating that CPExpert should produce Rule CIC330 whenever more than 75% of the queue index buffer pool buffers were used.

This finding is produced as an “early warning” of a potential problem. A large percent of LRU activity would be reported by **Rule CIC331**, and could imply “thrashing” in the queue index buffer pool.

You can alter this analysis using the **TSPCTFBP** guidance variable. For example, if you wish Rule CIC330 produced only when more than 90% of the queue index buffer pool buffers were used, specify:

```
%LET TSPCTFBP = 90 ; * PERCENT MINIMUM FREE BUFFERS IN BUFFER POOL;
```

Chapter 6.2.2: Percent LRU activity - TSPCTLRU variable

If a buffer is required from the LRU chain, this means that all other buffers were in use. If all other buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner. In either case, a large percent of LRU activity could imply “thrashing” in the queue index buffer pool.

CPExpert uses data in CICXQ2 to calculate the percent LRU activity. CPExpert produces Rule CIC331 when the percent LRU activity is greater than the value specified by the **TSPCTLRU** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRU** is 0.1, indicating that CPExpert should produce Rule CIC331 whenever the percent LRU activity is greater than 0.1% of the queue index buffer pool requests. This low percent LRU activity was set as a default to alert you to a potential problem that requires attention.

You can alter this analysis using the **TSPCTLRU** guidance variable. For example, if you wish Rule CIC331 produced only when the percent LRU activity is greater than 20% of the queue index buffer pool requests, specify:

```
%LET TSPCTLRU = 20 ; * PERCENT LRU ACTIVITY;
```


Chapter 6.2.3: Percent buffer pool buffers used- TSPCTUSE variable

The number of buffers in the queue index buffer pool is defined using the BUFFERS= keyword in the TS queue server parameters. The default specification is BUFFERS={100}, which specifies that 100 buffers should be allocated to for the server address space. IBM states that it is not worth defining extra buffers beyond the point where the definition might cause MVS paging, as it is more efficient to reread the index entry than to page in the buffer from auxiliary storage.

CPExpert uses data in CICXQ2 to calculate the maximum percent of the queue index buffer pool buffers that had been used.

CPExpert produces Rule CIC332 when the percent queue index buffer pool buffers used is less than the value specified by the **TSPCTUSE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTUSE** is 10, indicating that CPExpert should produce Rule CIC332 whenever less than 10% of the queue index buffer pool buffers were used. CPExpert suppresses this finding if 100 buffers or less are defined.

You can alter this analysis using the **TSPCTUSE** guidance variable. For example, if you wish Rule CIC331 produced only when less than 5% of the queue index buffer pool buffers were used, specify:

```
%LET TSPCTUSE = 5 ; * PERCENT BUFFERS USED IN INDEX BUFFER POOL;
```

Chapter 6.2.4: Percent wait on buffer lock - TSPCTWBL

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke *implicit enqueuing*. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

CICS provides two explicit enqueuing commands (EXEC CICS ENQ RESOURCE and EXEC CICS DEQ RESOURCE). These commands can be used to protect a temporary storage queue from being read and updated concurrently.

After a task has issued an ENQ RESOURCE(data-area) command, any other task that issues an ENQ RESOURCE command with the same data-area parameter is suspended until the task issues a matching DEQ RESOURCE(data-area) command, or until the UOW ends.

CPExpert uses data in CICXQ2 to calculate the percent of GET waits on buffer lock.

CPExpert produces Rule CIC334 when the percent GET waits on buffer lock is greater than the value specified by the **TSPCTWBL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWBL** is 1, indicating that CPExpert should produce Rule CIC334 whenever more than one percent of the GET requests waited for a buffer lock. CPExpert suppresses this finding unless there are more than 1,000 GET requests in the statistics interval.

You can alter this analysis using the **TSPCTWBL** guidance variable. For example, if you wish Rule CIC331 produced only when more than 5% of the GET requests waited for a buffer lock, specify:

```
%LET TSPCTWBL = 5 ; * PERCENT WAIT ON BUFFER LOCK;
```

Chapter 6.2.5: Percent wait on buffer pool lock - TSPCTWBP

Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region. There are two situations in which shared temporary storage requests can wait on the shared TS pool:

- **The queue pool is locked for exclusive use**. Message **DFHXQ0407** (CF structure strname is not available for shared use) issued if a request for a TS pool cannot be satisfied because the queue pool is locked for exclusive use by some other job (such as a queue pool unload or reload job).

  The server is terminated in this situation. Consequently, this situation is unlikely to be the cause of frequent waits on a shared temporary storage buffer pool lock.

- **SUSPEND on resource type TSPOOL**. Resource type TSPOOL indicates that the maximum number of concurrent requests (10) for a temporary storage pool in the coupling facility has been reached. The task resumes when one of the requests completes.

CPExpert uses data in CICXQ2 to calculate the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock.

CPExpert produces Rule CIC333 when the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock is greater than the value specified by the **TSPCTWPB** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWPB** is 0.1, indicating that CPExpert should produce Rule CIC333 whenever more than one tenth of the requests waited on a buffer pool lock.

You can alter this analysis using the **TSPCTWPB** guidance variable. For example, if you wish Rule CIC333 produced only when more than five percent of the requests waited on a buffer pool lock, specify:

```
%LET TSPCTWPB = 5 ; * PERCENT WAIT ON BUFFER POOL LOCK;
```

Chapter 6.3: Pool server storage statistics

Shared temporary storage queues are stored in named pools in an MVS coupling facility. Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a specific named pool. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. It also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage, for example sequential file read and write routines.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as the *AXMPGANY pool*. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as the *AXMPGLOW* pool.

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. If there is still insufficient storage to satisfy the request, the request fails.

Shared temporary storage queue server *storage statistics* are available in MXG file CICXQ3.

Chapter 6.3.1: AXMPGANY requests failed after retry - TSANYRQS |

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert produces Rule CIC341 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S3ANYRQS[16] in CICXQ3) is greater than the value | specified by the **TSANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **TSANYRQS** is 0, indicating that CPExpert should produce Rule CIC341 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **TSANYRQS** guidance variable. Since Rule CIC341 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the TSANYRQS guidance variable**. However, if you have very unusual circumstance[17] and you wish Rule CIC341 produced only when more than | five storage requests in the AXMPGANY pool failed after retry, specify:

```
%LET TSANYRQS = 5 ; * LOC=ANY REQUESTS FAILED AFTER RETRY;
```

Chapter 6.3.2: AXMPGLOW requests failed after retry - TSLOWRQS |

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert produces Rule CIC344 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S3LOWRQS[18] in CICXQ3) is greater | than the value specified by the **TSLOWRQS** guidance variable in USOURCE(CICGUIDE).

[16] Please note that the *CICS Performance Guides* describe S3ANYRQF as failed requests and S3ANYRQS as requests to release storage. These descriptions are "reversed" and IBM Hursley has agreed to correct the *CICS Performance Guides*. S3ANYRQF is the requests to "free" or release storage and S3ANYRQS is number of failed requests because of "short on storage".

[17] I would greatly appreciate being advised of any such circumstance, so I can alter CPExpert's logic and documentation accordingly. Please send a note to Don_Deese@cpexpert.com. Thanks!

[18] Please note that the *CICS Performance Guides* describe S3LOWRQF as failed requests and S3LOWRQS as requests to release storage. These descriptions are "reversed" and IBM Hursley has agreed to correct the *CICS Performance Guides*. S3LOWRQF is the requests to "free" or release storage and S3LOWRQS is number of failed requests because of "short on storage".

The default value for the **TSLOWRQS** is 0, indicating that CPExpert should produce Rule CIC344 when any storage request in the AXMPGLOW pool failed after retry.

You can alter this analysis using the **TSLOWRQS** guidance variable. Since Rule CIC344 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the TSLOWRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC341 produced only when more than five storage requests in the AXMPGLOW pool failed after retry, specify:

```
%LET TSLOWRQS = 5 ; * LOC=BELOW REQUESTS FAILED AFTER RETRY;
```

Chapter 6.3.3: Low percent AXMPGANY free storage - TSPCTAMN |

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGANY pool. |

CPExpert produces Rule CIC342 when the percent free storage in the AXMPGANY pool is less than the value specified by the **TSPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTAMN** is 25, indicating that CPExpert should produce Rule CIC342 whenever less than 25% of storage in the AXMPGANY pool is free.

You can alter this analysis using the **TSPCTAMN** guidance variable. For example, if you wish Rule CIC342 produced only when less than 25% of storage in the AXMPGANY pool is free , specify:

```
%LET TSPCTAMN = 5 ; * LOC=ANY PERCENT MINIMUM FREE STORAGE;
```

Chapter 6.3.4: Percent AXMPGANY requests retried - TSPCTARC |

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGANY storage POOL will correspondingly increase.

CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried.

CPExpert produces Rule CIC340 when the percent of storage requests in the AXMPGANY pool initially failed and were retried is greater than the value specified by the **TSPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTARC** is .1, indicating that CPExpert should produce Rule CIC340 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

You can alter this analysis using the **TSPCTARC** guidance variable. For example, if you wish Rule CIC340 produced only when more than five percent of storage requests in the AXMPGANY pool initially failed and were retried , specify:

```
%LET TSPCTARC = 5 ; * PERCENT LOC=ANY REQUEST FAILED AND RETRIED;
```

Chapter 6.3.5: Low percent AXMPGLOW free storage - TSPCTLMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGLOW pool.

CPExpert produces Rule CIC345 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **TSPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLMN** is 25, indicating that CPExpert should produce Rule CIC345 whenever less than 25% of storage in the AXMPGLOW pool is free.

You can alter this analysis using the **TSPCTLMN** guidance variable. For example, if you wish Rule CIC345 produced only when less than 25% of storage in the AXMPGLOW pool is free , specify:

```
%LET TSPCTLMN = 5 ; * LOC=LOW PERCENT MINIMUM FREE STORAGE;
```

Chapter 6.3.6: Percent AXMPGLOW requests retried - TSPCTLRC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the “compress attempt” to reduce fragmentation in the AXMPGLOW storage POOL will correspondingly increase.

CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried.

CPExpert produces Rule CIC340 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **TSPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRC** is .1, indicating that CPExpert should produce Rule CIC340 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

You can alter this analysis using the **TSPCTLRC** guidance variable. For example, if you wish Rule CIC340 produced only when more than five percent of storage requests in the AXMPGLOW pool initially failed and were retried , specify:

```
%LET TSPCTLRC = 5 ; * PERCENT LOC=LOW REQUEST FAILED AND RETRIED;
```


Chapter 6.4 Specifying guidance for specific shared TS pools

The guidance variables for shared temporary storage pools are globally applied during CPExpert's analysis of shared temporary storage performance from the perspective of CICS. These global guidance variables might not be applicable to some shared temporary storage pools, however. Guidance can **optionally** be applied to specific shared temporary storage pools <u>or to the coupling facility structures used by the shared temporary storage pools</u>.

Guidance for specific shared temporary storage pools is accomplished by specifying **%LET TSGUIDE=Y;** in USOURCE(CICGUIDE), identifying the shared temporary storage pools to which the guidance applies, and specifying guidance variables for the shared temporary storage pools.

Please note that the guidance is specified for specific shared temporary storage pools. The guidance is NOT specified for specific CICS regions, nor may guidance for shared temporary storage pools be "overridden" on a region-by-region basis. This is because shared temporary storage pools are not associated with a specific CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific shared temporary storage pools. As illustrated in Exhibit 3-4, guidance for individual shared temporary storage pools 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(CICGUIDE) member.

The shared temporary storage pools are identified by the POOL statement, which is used to specify the shared temporary storage pool to which the specific guidance applies.

Any number of shared temporary storage pools may be defined with appropriate guidance specified for each pool.

Following the POOL statement are the individual guidance statements for the shared temporary storage pool identified. Any of the global shared temporary storage pool guidance variables can be specified for the shared temporary storage pool.

If a particular shared temporary storage pool 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 G1POOL2 shared temporary storage pool server. In this example, 10 was specified as the guidance for the TSPCTLRU variable.

```
%LET TSGUIDE = Y;         * GUIDANCE IS PROVIDED FOR INDIVIDUAL SHARED TS POOLS;
/* SPECIFY GUIDANCE FOR SHARED TEMPORARY STORAGE POOLS
POOL  = G1POOL2
TSPCTLRU  =10  * PERCENT LRU ACTIVITY          ;
*/
```

Chapter 7: Shared Data Tables and CFDT Guidance Variables

For VSAM data sets, CICS file control provides applications with *file control commands* that read, update, add, and browse the data sets. There are many features provided by IBM that allow applications to use file control commands to reference VSAM data sets, and allow separate applications to *share* the data. Two such features are the *shared data table* approach and the Coupling Facility Data Table approach.

- **Shared data tables**. An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

  With shared data table support, the KSDS file is called the *source data set*. The copy of the file in memory is called the *data table*. The process of copying the records from the file to the data table is called *loading the data table*. Whenever a CICS application wishes to reference the VSAM file using normal file control commands, CICS attempts to use the representation of the file in the data table, rather than accessing the source data.

- **Coupling Facility Data Tables.** CICS Coupling Facility Data Tables provide a significant enhancement to shared data tables in a parallel sysplex. The CFDT design provides an excellent way to share file data using CICS file control, without resorting to VSAM record level sharing (RLS). The CFDT design eliminates the requirement for having a File Owning Region (as is required with normal shared data tables).

  CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

The CICS Component anlayzes data in CICCFS6D, CICCFS7D, CICCFS8D, and CICCFS9D to detect performance problems or potential performance problems with coupling facility data tables.

Exhibit 3-4 illustrates the USOURCE(CICGUIDE) variables that provide guidance to the CICS Component as it analyzes shared data tables and coupling facility data tables. This chapter describes the guidance variables that can be used to alter CPExpert's analysis of shared data tables or coupling facility data tables.


```
*********************************************************************    ;
*      CICS Component GUIDANCE VARIABLES                                 ;
*********************************************************************    ;
      •
      •
      •
      •

* SHARED DATA TABLE/COUPLING FACILITY DATA TABLE GUIDANCE VARIABLES ;
*                                                                   ;
%LET CICSRNF  = 100  ;  * ACCEPTABLE RECORDS NOT FOUND, CICS-SDT    ;
%LET MINSDTIO = 500  ;  * MINIMUM VSAM DATA TABLE IO (RULE 405/406) ;
%LET PCTDTSRC = 25   ;  * PCT SDT ACCESSES TO SOURCE DATA SET       ;
%LET SDTFULL  = 1    ;  * NUMBER SHARED DATA TABLE FULL CONDITIONS  ;
%LET UMTRNF   = 0    ;  * ACCEPTABLE RECORDS NOT FOUND, UMT-SDT     ;
* CFDT SERVER COUPLING FACILITY STATISTICS GUIDANCE;
%LET CFLSTFUL = 0    ;  * TIMES MAXNUMRECS WAS REACHED FOR CFDT     ;
%LET CFNOSPCE = 0    ;  * TIMES LIST STRUCTURE WAS OUT OF SPACE     ;
%LET CFPCTELE = 70   ;  * MAXIMUM PERCENT DATA ELEMENTS USED        ;
%LET CFPCTENT = 70   ;  * MAXIMUM PERCENT LIST ENTRIES USED         ;
%LET CFPCTRNF = .1   ;  * ACCEPTABLE PCT RECORDS NOT FOUND, CFDT    ;
* CFDT POOL SERVER STORAGE STATISTICS GUIDANCE;
%LET CFANYRQS = 0    ;  * LOC=ANY STORAGE REQUEST FAILED AFTER RETRY;
%LET CFLOWRQS = 0    ;  * LOC=LOW STORAGE REQUEST FAILED AFTER RETRY;
%LET CFPCTAMN = 25   ;  * LOC=ANY PERCENT MINIMUM FREE STORAGE      ;
%LET CFPCTARC = .1   ;  * LOC=ANY STORAGE REQUEST FAILED AND RETRIED;
%LET CFPCTLMN = 25   ;  * LOC=LOW PERCENT MINIMUM FREE STORAGE      ;
%LET CFPCTLRC = .1   ;  * LOC=LOW STORAGE REQUEST FAILED AND RETRIED;
********************************************************************;
```

DEFAULT VALUES FOR SHARED TEMPORARY STORAGE ANALYSIS

EXHIBIT 3-4

Guidance variables for shared data tables and coupling facility data tables are described in three parts: (1) shared data table statistics guidance variables, (2) CFDT pool server coupling facility statistics guidance variables, and (3) CFDT pool server storage statistics guidance variables.

Chapter 7.1: Shared data table statistics guidance variables

The CICS Component analyzes data in CICFCR to detect performance problems or potential performance problems with shared data tables. The following paragraphs describe the guidance variables that apply to shared data tables.

Chapter 7.1.1: Records not found in CICS-maintained shared data table - CICSRNF

CICS supports two types of data table: (1) CICS-maintained data tables and (2) user-maintained data tables. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is automatically applied to the source data set *before* being applied to the data table. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Applications can reference the data table during initial loading. These references will produce a “record not found” condition if the references are to records outside the range of those already loaded into the data table. For CICS-maintained data tables, small numbers of the “record not found” condition after initial loading should not normally be a cause for alarm. However, a large number of “records not found” will cause unnecessary overhead and delay to applications.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of times records were not found for CICS-maintained data tables. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data table; CICS was required to retrieve the record from the source data set.

CPExpert produces Rule CIC402 when the A17DTRNF value is greater than the **CICSRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CICSRNF** guidance variable is 100, indicating that CPExpert should produce Rule CIC402 when more than one hundred read requests resulted in a “record not found” condition.

You can change the **CICSRNF** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC402 is produced too often. For example, if you wish Rule CIC402 produced only when CICS encountered a “record not found” condition more than 500 times, specify:

```
%LET CICSRNF = 500 ; * ACCEPTABLE RECORDS NOT FOUND;
```


Chapter 7.1.2: Minimum shared data table I/O - MINSDTIO

An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. Any operation requiring access to the source data set reduces the efficiency of the data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data for a CICS-maintained shared data table. Based on this calculated percent, CPExpert will produce Rule CIC405 (high data set activity for CICS-maintained shared data table) or Rule CIC406 (VSAM data set might not be good candidate or shared data table).

Before producing Rule CIC405 or Rule CIC406, however, CPExpert applies the **MINSDTIO** variable to ensure that findings are produced only for shared data tables with a reasonably high amount to of I/O activity. The default value for the **MINSDTIO** is 500, indicating that Rule CIC405 and Rule CIC406 will be suppressed unless at least 500 VSAM file control commands are issued to the shared data table.

You can change the **MINSDTIO** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC405 or Rule CIC406 are produced too often. For example, if you wish this analysis to be applied to shared data tables only when more than 1000 file control requests were to a shared data table, specify:

```
%LET MINSDTIO = 1000 ; * MINIMUM VSAM I/O ACTIVITY (CIC405/CIC406)
```

Chapter 7.1.3: Percent data table access to source data - PCTDTSRC

An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. Any operation requiring access to the source data set reduces the efficiency of the data table.

If a large percent of file control commands access the VSAM source data set, the benefits of using a data table can outweigh the overhead and virtual storage costs of maintaining a data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data for a CICS-maintained shared data table.

CPExpert produces Rule CIC405 when the percent VSAM source data set accesses is more than the value specified by the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTDTSRC** is 50 indicating that CPExpert should produce Rule CIC405 whenever more than 50% of the file accesses required that CICS access the VSAM source data set.

You can change the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC405 is produced too often. For example, if you wish this analysis to be applied to shared data tables only when more than 75% of the file control requests to a shared data table went to the VSAM source data set, specify:

```
%LET PCTDTSRC = 75% ; * PERCENT SDT REQUESTS TO VSAM SOURCE DATA SET
```

Chapter 7.1.4: Number of times shared data table was full - SDTFULL

Records are placed into a shared data table in one of three ways: (1) records placed in the data table by the initial loading of the data table from the source data set, (2) records subsequently added to the data table from the source data set, and (3) new records written to the data table after the data table has been loaded.

Regardless of how records are added to a shared data table, the number of records in the table cannot exceed the number specified by the MAXNUMRECS variable specified for the table. If that limit is reached during loading or while adding to the table, a “table full” condition applies. Encountering a “table full” condition can have several undesirable effects.

CPExpert uses the A17DTATF variable in the MXG CICFCR file to assess the number of records that CICS attempted to add to the table but was unable to do so because the table was full. The A17DTATF value means that the data table already contained the maximum number of records specified in the MAXNUMRECS parameter of the DEFINE FILE command.

CPExpert produces Rule CIC401 when the A17DTATF value is greater than the **SDTFULL** guidance variable in USOURCE(CICGUIDE). The default value for the **SDTFULL** guidance variable is one, indicating that CPExpert should produce Rule CIC401 when more than one shared data table full condition was detected. Note that the initial loading of the shared data table normally will produce a count of one for the A17DTATF, so the default value of the SDTFULL variable accounts for this normal situation.

You can change the **STDFULL** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC401 is produced too often. For example, if you wish Rule CIC401 produced only when a shared data table was full more than 100 times, specify:

```
%LET SDTFULL = 100 ; * TIMES SHARED DATA TABLE WAS FULL;
```

Chapter 7.1.5: Records not found in user-maintained shared data table - UMTRNF

CICS supports two types of data table: (1) CICS-maintained data tables and (2) user-maintained data tables. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is automatically applied to the source data set *before* being applied to the data table. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Applications can reference the data table during initial loading. These references will produce a “record not found” condition if the references are to records outside the range of those already loaded into the data table. For user-maintained data tables, the “record not found” condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the data table, any “record not found” condition for user-maintained data tables normally indicates a coding or logic error with the application.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of times records were not found for usesr-maintained data tables. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data tablet.

CPExpert produces Rule CIC403 when the A17DTRNF value is greater than the **UMTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **UMTRNF** guidance variable is 0, indicating that CPExpert should produce Rule CIC403 when any read request resulted in a “record not found” condition.

You can change the **UMTRNF** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC403 is produced too often. For example, if you wish Rule CIC403 produced only when CICS encountered a “record not found” condition more than 100 times for a user-maintained data table, specify:

```
%LET UMTRNF = 100 ; * ACCEPTABLE RECORDS NOT FOUND;
```

Chapter 7.2: CFDT pool server coupling facility structure statistics

A coupling facility data table is contained in a *named CFDT pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility. CFDT pool server *coupling facility statistics* are available in MXG file CICCFS6D.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With coupling facility data tables, the list structure is the named CFDT pool, while the lists themselves are CFDTs within the named pool. The CFDT pool server designates the maximum number of CFDTs the CFDT pool is to have, and allocates the list structure based on parameters that are provided to the CFDT pool server when the pool server region is started.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both. Data entries are composed of units of storage called *data elements*. Although a data entry can be composed of a number of data elements, list operations treat the data entry as a single entity; data elements cannot be read or written individually.

Chapter 7.2.1: Times CFDT was full - CFLSTFUL variable

The MAXNUMRECS parameter provided with the VSAM file definition specifies the maximum number of records that can be in a coupling facility data table. The default value for the MAXNUMRECS parameter is NOLIMIT, indicating that there is no limit on the maximum number of records. When the number of records in the CFDT reaches the MAXNUMRECS value, the CFDT is marked “full” and no further records can be added to the CFDT until records are deleted to make space. The WRITE request is rejected, and the transaction must handle the NOSPACE return with an exception handling routine.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP6CT variable contains a count of the number of times that a list (or table) became full. CPExpert calculates the percent of requests that encountered a "list full" condition.

CPExpert produces Rule CIC424 when the number of List Full conditions is greater than the value specified by the **CFPCTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTFUL** is 0, indicating that CPExpert should produce Rule CIC424 whenever any List Full conditions occurred.

You can alter this analysis using the **CFLSTFUL** guidance variable. For example, if you wish Rule CIC424 produced only when more than one List Full condition occurred, specify:

```
%LET CFLSTFUL = 1 ; * TIMES LIST WAS FULL;
```

Chapter 7.1.2: List structure was out of space - CFNOSPCE variable

Coupling Facility Data Tables are kept in a *named pool* in an MVS coupling facility. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a *list structure* in a coupling facility.

If the CFDT list structure is allowed to become completely full,message **DFHCF0442** *(CF structure strname request failed, structure is full*) is issued and CICS rejects any attempt to add new records to a CFDT or to create new tables in the pool. Additionally, completely filling a CFDT list structure can have a significant impact on performance and application function. IBM

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to determine whether a Structure Full condition occurred for a CFDT pool server list structure. CICCFS6D variable S6RSP7CT (List structure became full) indicates that the list structure was full.

CPExpert produces Rule CIC425 when the number of Structure Full conditions is greater than the value specified by the **CFNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFNOSPCE** is 0, indicating that CPExpert should produce Rule CIC425 whenever any Structure Full conditions occurred.

You can alter this analysis using the **CFNOSPCE** guidance variable. For example, if you wish Rule CIC324 produced only when more than one Structure Full condition occurred, specify:

```
%LET CFNOSPCE = 1 ; * TIMES LIST STRUCTURE WAS OUT OF SPACE;
```

Unless you have a very unique situation, you should **not** alter the CFNOSPCE guidance variable. You should always be aware of Structure Full situations.

Chapter 7.1.3: Percent data elements in use - CFPCTELE variable

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

IBM suggests that no more than 75% of the list structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the structure list elements that had been used.

CPExpert produces Rule CIC420 when the maximum percent structure list elements used is more than the value specified by the **CFPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTELE** is 70 indicating that CPExpert should produce Rule CIC420 whenever more than 70% of the list entries had been used.

You can alter this analysis using the **CFPCTELE** guidance variable. For example, if you wish Rule CIC420 produced only when more than 60% of the data elements had been used, specify:

```
%LET CFPCTELE = 60 ; * MAXIMUM PERCENT DATA ELEMENTS USED;
```

Chapter 7.1.4: Percent list entries in use - CFPCTENT variable

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

IBM suggests that no more than 75% of the list structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used. CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the structure list elements that had been used.

CPExpert produces Rule CIC421 when the maximum percent data entries used is more than the value specified by the **CFPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTENT** is 70 indicating that CPExpert should produce Rule CIC421 whenever more than 70% of the data entries had been used.

You can alter this analysis using the **CFPCTENT** guidance variable. For example, if you wish Rule CIC421 produced only when more than 60% of the list entries had been used, specify:

```
%LET CFPCTENT = 60 ; * MAXIMUM PERCENT LIST ENTRIES USED;
```

Chapter 7.1.5: Percent record not found - CFPCTRNF variable

Applications access data in a CFDT using standard file control commands (read, write, delete, etc.). Records are placed into a CFDT in one of two ways: (1) records placed in the CFDT by the initial loading of the CFDT from the source data set (if a source data set is defined), and (2) new records written to the CFDT after the CFDT has been loaded.

Applications can reference the CFDT during initial loading. These references will produce a “record not found” condition if the references are to records outside the range of those already loaded into the data table. These conditions should be small and can normally be ignored.

The “record not found” condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the CFDT (if a source data set existed), any “record not found” condition for normally indicates (1) a coding or logic error with the application, (2) an attempt to reference a record that should be in the CFDT (but was not in the CFDT because, for example, the MAXNUMRECS had been reached or the record had been suppressed by one of the user exits mentioned above), or (3) a deliberate reference to determine whether the record exists.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP3CT variable contains a count of the number of times CICS attempted to read a record from the CFDT, but was unable to satisfy the read request because the record was not in the CFDT. CPExpert calculates the percent of requests that encountered a “specified entry (table or item) was not found” condition.

CPExpert produces Rule CIC423 when the percent requests that resulted in a “entry (table or item) not found” condition is greater than the **CFPCTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTRNF** guidance variable is 0.1, indicating that CPExpert should produce Rule CIC423 when more than 0.1% of the requests resulted in a “entry (table or item) not found” condition.

You can alter this analysis using the **CFPCTRNF** guidance variable. For example, if you wish Rule CIC423 produced any file control requests to a CFDT encountered a “record not found” condition, specify:

```
%LET CFPCTNOE = 0 ; * PERCENT RECORDS NOT FOUND IN CFDT;
```

Chapter 7.3: Server buffer pool statistics

A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for each CFDT pool defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool.

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. If there is still insufficient storage to satisfy the request, the request fails.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D.

Chapter 7.3.1: AXMPGANY requests failed after retry - CFANYRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC441 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S9ANYRQS[19]) is greater than the value specified by the **CFANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFANYRQS** is 0, indicating that CPExpert should produce Rule CIC441 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **CFANYRQS** guidance variable. Since Rule CIC441 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFANYRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC441 produced only when more than five storage requests in the AXMPGANY pool failed after retry, specify:

```
%LET CFANYRQS = 5 ; * LOC=ANY REQUESTS FAILED AFTER RETRY;
```

Chapter 7.3.2: AXMPGLOW requests failed after retry - CFLOWRQS

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC444 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S9LOWRQS[20]) is greater than the value

[19] Please note that the *CICS Performance Guides* describe S9ANYRQF as failed requests and S9ANYRQS as requests to release storage. These descriptions are “reversed” initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9ANYRQF is the requests to “free” or release storage and S9ANYRQS is number of failed requests because of “short on storage”.

[20] Please note that the *CICS Performance Guides* describe S9LOWRQF as failed requests and S9LOWRQS as requests to release storage. These descriptions are “reversed” initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9ANYRQF is the requests to “free” or release storage and S9ANYRQS is number of failed requests because of “short on storage”.

specified by the **CFLOWRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFLOWRQS** is 0, indicating that CPExpert should produce Rule CIC444 when any storage request in the AXMPGANY pool failed after retry.

You can alter this analysis using the **CFLOWRQS** guidance variable. Since Rule CIC441 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFLOWRQS guidance variable**. However, if you have very unusual circumstance and you wish Rule CIC444 produced only when more than five storage requests in the AXMPGLOW pool failed after retry, specify:

```
%LET CFLOWRQS = 5 ; * LOC=LOW REQUESTS FAILED AFTER RETRY;
```

Chapter 7.3.3: Low percent AXMPGANY free storage - CFPCTAMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICCFS9D to calculate the minimum percent of free storage in the AXMPGANY pool.

CPExpert produces Rule CIC442 when the percent free storage in the AXMPGANY pool is less than the value specified by the **CFPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTAMN** is 25, indicating that CPExpert should produce Rule CIC442 whenever less than 25% of storage in the AXMPGANY pool is free.

You can alter this analysis using the **CFPCTAMN** guidance variable. For example, if you wish Rule CIC442 produced only when less than 25% of storage in the AXMPGANY pool is free, specify:

```
%LET CFPCTAMN = 5 ; * LOC=ANY PERCENT MINIMUM FREE STORAGE;
```

Chapter 7.3.4: AXMPGANY requests retried - CFPCTARC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGANY storage POOL will correspondingly increase.

CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried.

CPExpert produces Rule CIC440 when the percent of storage requests in the AXMPGANY pool initially failed and were retried is greater than the value specified by the **CFPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTARC** is .1, indicating that CPExpert should produce Rule CIC440 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

You can alter this analysis using the **CFPCTARC** guidance variable. For example, if you wish Rule CIC440 produced only when more than five percent of storage requests in the AXMPGANY pool initially failed and were retried , specify:

```
%LET CFPCTARC = 5 ; * PERCENT LOC=ANY REQUEST FAILED AND RETRIED;
```

Chapter 7.3.5: Low percent AXMPGLOW free storage - CFPCTLMN

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICFS9D to calculate the minimum percent of free storage in the AXMPGLOW pool.

CPExpert produces Rule CIC445 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **CFPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLMN** is 25, indicating that CPExpert should produce Rule CIC445 whenever less than 25% of storage in the AXMPGLOW pool is free.

You can alter this analysis using the **CFPCTLMN** guidance variable. For example, if you wish Rule CIC445 produced only when less than 25% of storage in the AXMPGLOW pool is free, specify:

```
%LET CFPCTLMN = 5 ; * LOC=LOW PERCENT MINIMUM FREE STORAGE;
```

Chapter 7.3.6: AXMPGANY requests retried - CFPCTLRC

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the “compress attempt” to reduce fragmentation in the AXMPGLOW storage POOL will correspondingly increase.

CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried. CPExpert produces Rule CIC443 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **CFPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLRC** is .1, indicating that CPExpert should produce Rule CIC443 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

You can alter this analysis using the **CFPCTLRC** guidance variable. For example, if you wish Rule CIC443 produced only when more than five percent of storage requests in the AXMPGLOW pool initially failed and were retried , specify:

```
%LET CFPCTLRC = 5 ; * PERCENT LOC=LOW REQUEST FAILED AND RETRIED;
```

Chapter 7.4 Specifying guidance for specific shared data tables

The guidance variables for shared data tables (including Coupling Facility Data Tables) are globally applied during CPExpert's analysis of shared data table performance. These global guidance variables might not be applicable to some shared data tables or Coupling Facility Data Tables, however. Guidance can **optionally** be applied to specific shared data tables or to specific Coupling Facility Data Table (CFDT) servers <u>(and to the coupling facility structures used by the CFDT servers</u>.

Guidance for specific shared data tables (or for CFDT servers) is accomplished by specifying **%LET DTGUIDE=Y;** in USOURCE(CICGUIDE), identifying the shared data table names (or CFDT server names) to which the guidance applies, and specifying guidance variables for the shared data tables or CFDT servers.

Please note that the guidance is specified for specific shared data tables (or CFDT servers). The guidance is NOT specified for specific CICS regions, nor may guidance for shared data tables be "overridden" on a region-by-region basis. This is because shared data tables are not associated with a specific CICS region (even though a CICS region will act as a File Owning Region for the shared data table), and because a CFDT server is not associated with a CICS region.

Exhibit 3-4 illustrates the portion of CPEXPERT.USOURCE(CICGUIDE) that contains the analysis guidance variables for specific shared data tables or CFDT server. As illustrated in Exhibit 3-4, guidance 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(CICGUIDE) member.

The shared data tables (or CFDT servers) are identified by the TABLE statement, which is used to specify the shared data table (or CFDT server) to which the specific guidance applies.

Any number of shared data tables (or CFDT servers) may be defined with appropriate guidance specified for each tablel.

Following the TABLE statement are the individual guidance statements for the shared data table (or CFDT server) identified. Any of the global shared data table (or CFDT) guidance variables can be specified for the shared data table (or CFDT).

If a particular shared data table 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 G1POOL2 shared temporary storage pool server. In this example, 10 was specified as the guidance for the TSPCTLRU variable.

```
%LET TSGUIDE = Y;        * GUIDANCE IS PROVIDED FOR INDIVIDUAL SHARED TS POOLS;
/* SPECIFY GUIDANCE FOR SHARED TEMPORARY STORAGE POOLS
POOL  = G1POOL2
TSPCTLRU  =10  * PERCENT LRU ACTIVITY          ;
*/
```

Section 4: Executing the CICS Component

This section describes how to execute the CICS Component of CPExpert.

The instructions in this section assume that you have installed the CPExpert software. The CICS Component is installed as normal part of installing CPExpert, and the installation steps unique to the CICS Component are described in Section 2 of this User Manual. If you have not installed CPExpert, please install the software and implement Section 2 of this manual before continuing.

Step 1. Use TSO ISPF to create the Job Control Language

Use TSO ISPF to create the Job Control Language (JCL) necessary to execute the CICCPE code. Exhibit 4-1 illustrates sample JCL statements. These sample statements are available in CPExpert SOURCE(CICJCL).

```
//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.CPEXPERTSOURCE,DISP=SHR
//CPEDATA     DD     DSN=prefix.CPEXPERT.CPEDATA,DISP=OLD
//PDBLIB      DD     DSN=prefix.MXG.PDB,DISP=SHR
//CICLIB      DD     DSN=prefix.MXG.CICSPDB,DISP=SHR
//SYSIN       DD     DSN=prefix.CPEXPERT.SOURCE(CICCPE),DISP=SHR
```

JOB CONTROL LANGUAGE TO EXECUTE THE CICCPE MODULE

EXHIBIT 4-1

- The **SOURCE** and **USOURCE** DD statements in Exhibit 4-1 refer to the CPEXPERT.SOURCE and CPEXPERT.USOURCE, respectively.

- The **CPEDATA** DD statement in Exhibit 4-1 refers to the SAS data library maintained by CPExpert. The space for this library was created during the installation of CPExpert.

- The **PDBLIB** DD statement in Exhibit 4-1 refers to the SAS library containing the SMF/RMF performance data base to be analyzed. The example shows a sample DSN for a typical MXG performance data base.

- The **CICLIB** DD statement in Exhibit 4-1 refers to the SAS library containing the CICS performance data base to be analyzed. The example shows a sample DSN for a typical MXG performance data base. The CICLIB DD statement could (and often would) point to the same data set as the PDBLIB statement.

Step 2: Make any appropriate changes to the CICGUIDE Module

Before submitting the JCL shown in Exhibit 4-1 and executing the CICCPE Module, you should make appropriate changes to the CPEXPERT.USOURCE(CICGUIDE) module. These changes are described in Section 3 of this manual.

Step 3. Execute the CICCPE Module

Submit the JCL to execute the CICCPE Module. Most installations execute the CICCPE Module on a daily basis, after their normal update of their performance data base.

Checklist for Executing the CICS Component, Mainframe

☐ Make any necessary changes to USOURCE(CICGUIDE).

☐ Change the data set names in the sample DD statements.

☐ Submit the JCL to execute the CICCPE Module.

Section 5: Using the CICS Component

This section describes how to use the CICS Component to analyze constraints to improved performance of a CICS region.

As discussed in Section 1, the main purpose of the CICS Component of CPExpert is to evaluate the performance of an individual CICS region, identify potential constraints to improved performance, and suggest ways to eliminate the constraints.

This process is described in terms of providing guidance to the CICS Component, actions that should be taken on a daily basis, and actions that should be taken on a monthly or periodic basis.

Chapter 1: Prepare guidance for the CICS Component

Use TSO ISPF to change the CPEXPERT.USOURCE(CICGUIDE) PDS member to reflect the guidance required by the CICS Component. Exhibit 3-1 illustrates the variables that must be changed in the CICGUIDE PDS member.

This step must be taken only when the guidance changes. Do not hesitate to change the guidance if you feel that the rules are applying inappropriate analysis or if there are areas in which you do not wish to make a change.

Chapter 2: Actions on a daily basis

Use the CICS Component to a daily analysis of constraints to improved performance in the CICS region, by following the below steps:

Step 1: Execute the CICCPE Module

The JCL to execute the CICCPE module is described in Section 4. We normally suggest that you append the JCL to execute the CICCPE module to the normal daily update of your performance data base.

Step 2: Review the output from the CICCPE Module

If any rules were produced, refer to the specific rule in Appendix A for a description of the rule, a discussion of why the rule was produced, and suggestions 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 data, or you may wish to wait until enough information has been placed the historical files maintained by the CICS Component that a historical analysis can be performed.

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

- The CICS 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 CICS 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.

NOTE: 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 can be modified to help you analyze a problem. We will improve our product and you will have more help!

Description of Rules

Appendix A

This appendix contains a description of each rule that results in a finding by the CICS Component of CPExpert. The description summarizes the rule, lists predecessor rules, discusses the rationale for the finding, and suggests action.

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

If appropriate, 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 CICS and its interaction with the computer system (the hardware, MVS, 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. Additionally, the discussion generally explains how CPExpert determines whether the rule is appropriate to potentially improving the performance of the CICS region being analyzed.

The recommendations suggest 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. The LIST OF RULES in this appendix lists all rules that are included in the initial release of the CICS Component.

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 CICCPE Module.

Beginning with CICS/ESA Version 4.1, the references to IBM documents **apply to IBM BookManager 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.

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

RULE	DESCRIPTION
CIC101	The CICS region reached maximum tasks too often
CIC102	The AMAX value may be too low
CIC103	Class maximum was reached too often
CIC104	Maximum Task (MXT) specification may be too large
CIC105	Transaction class reached MAXACTIVE too often
CIC106	Too many tasks were queued for MAXACTIVE reason
CIC107	PURGETHRESH value should be increased
CIC108	Maximum Task (MXT) specification may be too small
CIC109	CICS region is approaching maximum capacity
CIC110	CICS encountered a Short-on-Storage condition
CIC112	The number of GETMAINs may be too high
CIC113	Peak EDSA usage is approaching EDSALIM value
CIC120	A storage violation occurred in a production CICS region
CIC121	Transactions experiencing storage violations
CIC130	Storage dumps occurred in a production CICS region
CIC140	The number of transaction errors is high
CIC141	Transactions experienced a stall-purge

RULE	DESCRIPTION
CIC150	VTAM reached the maximum RAPOOL value too often
CIC151	The RAPOOL value may be too large
CIC152	The VTAM short on storage count is increasing
CIC155	The number of concurrent Autoinstall attempts is high
CIC156	The DSHIPINT value might be too large
CIC160	The number of strings may be too low for VSAM NSR file
CIC161	The number of strings is too low for VSAM LSR pool
CIC162	The number of strings may be to high for VSAM LSR pool
CIC163	The number of buffers is too low for VSAM LSR pool
CIC164	The number of buffers may be to high for VSAM LSR pool
CIC165	The "look-aside" read hit ratio was low for VSAM LSR pool
CIC166	VSAM LSR files have same CI size for data and index records
CIC167	Too few VSAM files are assigned to LSR pools
CIC168	LSR buffer sizes may be inappropriate
CIC169	The number of buffers may be too low for VSAM LSR pool
CIC170	More than one string specified for write-only ESDS file
CIC171	Significant percent of write activity for ESDS file
CIC175	VSAM LSR pool was seldom used
CIC176	VSAM LSR subpool was seldom used
CIC177	Large percent UPDATE option used without WRITE or DELETE

RULE	DESCRIPTION
CIC184	Size of the temporary storage data set may be too small
CIC185	The number of Temporary Storage buffers may be too high
CIC186	The number of Temporary Storage buffers may be too low
CIC187	The number of Temporary Storage strings may be too high
CIC188	The number of Temporary Storage strings may be too low
CIC190	The number of DL/I Threads may be too low
CIC191	The DLTHRED specification may be too high
CIC192	The DMBPL specification may be too low
CIC193	The ENQPL specification may be too low
CIC194	The PSBPL specification may be too low
CIC210	The number of Transient Data buffers may be too low
CIC211	The number of Transient Data strings may be too low
CIC220	The CICS journal buffer size is too small (system log)
CIC221	The CICS journal buffer size may be too small (user log)
CIC222	The CICS journal buffer size may be too small (shift-ups)
CIC230	The CICS Dynamic Transaction Backout buffer size may be too small

RULE	**DESCRIPTION**
CIC250	The ICV value may be inappropriate
CIC251	Default value was specified for the MROBTCH parameter
CIC252	The MROBTCH value may be too large
CIC253	Default value was specified for the MROLRM parameter
CIC254	ISRDELAY value may be too low
CIC255	PVDELAY value may be too low
CIC256	CICS Monitoring Facility (Performance Class) was turned on
CIC257	CICS Auxiliary Trace was turned on
CIC258	CICS Internal Trace Table might be too small
CIC259	SUBTSKS=1 was specified in System Initialzation Table
CIC260	Insufficient sessions may have been defined
CIC261	APPC modegroup may have insufficient sessions defined
CIC262	Conflicting usage between APPC modegroups may exist
CIC263	Balance between contention winners and contention losers may be inappropriate
CIC264	Balance between contention winners and contention losers may be inappropriate for APPC modegroup
CIC265	Connection problems caused generic ALLOCATE requests to fail
CIC266	Connection problems caused specific ALLOCATE requests to fail
CIC267	Insufficient sessions may have been defined
CIC275	CICS-DB2 peak TCBs is approaching TCBLIMIT
CIC276	High number of CICS-DB2 peak tasks on Pool Ready Queue
CIC277	CICS-DB2 pool threads in use is approaching limit

RULE	DESCRIPTION
CIC301	CICS waited for logger “structure full” condition
CIC302	CICS waited for logger “staging data set full” condition
CIC303	Log stream structure offloads occurred: 90% full
CIC304	Interim storage was not efficiently used for log stream
CIC305	Local storage buffers not efficiently used, DASD-only log stream
CIC306	DASD staging data set high threshold was reached
CIC307	Frequent log stream DASD-shifts occurred
CIC308	Log stream caused structure to reach high threshold
CIC309	Log stream consumed most of structure resources
CIC320	High percent of structure entries were in use
CIC321	High percent of structure elements were in use
CIC322	High percent structure requests timed out by coupling facility
CIC323	High percent entries (queue or item) not found
CIC324	Maximum list key was reached
CIC325	The list structure was out of space
CIC326	High percent of list data reads had to be repeated
CIC327	High percent of structure index data reads had to be repeated
CIC328	High percent version check failed for an entry being updated
CIC330	High percent shared TS queue index buffers were in use
CIC331	High percent LRU activity in the TS queue index buffer pool
CIC332	Excess buffers defined for queue index buffer pool
CIC333	High percent waits on shared temporary storage buffer pool lock

RULE	DESCRIPTION
CIC334	High percent GET wait on shared temporary storage buffer lock
CIC340	LOC=ANY storage request initially failed and was retried
CIC341	LOC=ANY storage request was unable to obtain requested storage after retry
CIC342	LOC=ANY low percent minimum free storage
CIC343	LOC=BELOW storage request initially failed and was retried
CIC344	LOC=BELOW storage request was unable to obtain requested storage after retry
CIC345	LOC=BELOW low percent minimum free storage

RULE	DESCRIPTION
CIC401	Adds were rejected because shared data table was full
CIC402	Records not found in CICS-maintained shared data table
CIC403	Records not found in user-maintained shared data table
CIC405	High data set activity for CICS-maintained shared data table
CIC406	VSAM data set might not be good candidate for shared data table
CIC420	High percent of structure entries were in use
CIC421	High percent of structure elements were in use
CIC422	High percent structure requests encountered length error
CIC423	High percent entries (CFDT or item) not found
CIC424	List full condition was reached for CFDT
CIC425	The CFDT list structure was out of space
CIC426	High percent version check failed for an entry being updated
CIC440	LOC=ANY storage request initially failed and was retried
CIC441	LOC=ANY requests were unable to obtain storage and failed
CIC442	LOC=ANY low percent minimum free storage
CIC443	LOC=BELOW storage request initially failed and was retried
CIC444	LOC=BELOW requests were unable to obtain storage and failed
CIC445	LOC=BELOW low percent minimum free storage
CIC446	CFDT pool server storage allocation was less than expected

Rule CIC101: The CICS region reached MAX tasks too often

Finding: CPExpert has detected that the CICS region reached maximum tasks (MXT value) too often.

Impact: This finding has a MEDIUM IMPACT on the performance of the CICS region. If this is a MRO environment, the finding can have a HIGH IMPACT on performance, since CICS can become deadlocked if MXT limits AOR/TOR/FOR regions.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. For CICS versions prior to Version 1.7, CICS stops inviting input messages when the number of tasks concurrently in the region reaches the MXT value.

Beginning with CICS Version 1.7, CICS will not create a Task Control Area (TCA) for a new task when the number of tasks concurrently in the region reaches the MXT value.

The MXT operand is used to limit the resources used by CICS tasks. There are two general reasons for restricting the number of concurrent tasks:

- CICS is becoming short on storage, encountering a stress condition, and overall performance suffers. Under these circumstances, it may be better to limit the number of concurrent tasks so that CICS does not enter a stress condition. This limit may be necessary until actions are taken to relieve storage constraint or until additional resources are acquired.

- Processor resources are significantly constrained and management has decided to limit the processor demands of CICS. MXT can be used to limit the size of the CICS active and suspended task chains and minimize the processor resources used in task control.

The MXT operand normally should be used only to restrict the use of storage resources (the AMXT operand can be used to limit the number of tasks using other resources, such as processor resources). If storage is not a constraint, the MXT operand generally should not limit access to CICS.

CICS performance can be unnecessarily degraded if the MXT value is used to restrict the number of concurrent tasks, and if the system is capable of handling more tasks.

Additionally, CICS deadlocks can occur if the MXT operand is set too low. This particularly can occur in a MRO environment. If a slowdown occurs for some reason, tasks can become backed up in one or more of the AOR/TOR/FOR regions and CICS can enter a deadlock.

Some installations have specified MXT=999, with the intention of allowing an unlimited number of tasks in storage. In fact, this specification is recommended by IBM's CICS MOR Tuning and Performance Guide. This specification can have an adverse effect on CICS performance for CICS Version 3 (see Rule CIC104).

CPExpert produces Rule CIC101 if the CICS statistics reported that the maximum tasks value was reached more than the MAXTASK guidance variable and CPExpert did not detect a storage constraint. The default for the MAXTASK guidance variable is zero.

Suggestion: CPExpert suggests that you consider increasing the MXT value in the SIT. The value for the MXT operand normally should be sufficiently high that tasks are not restricted by MXT. Of course, if storage is a constraint, you might have to use the MXT operand to restrict access to CICS by tasks. However, be very careful not to allow the MXT value to be reached in a MRO environment (unless you have a very unusual environment).

IBM suggests that an effective way of finding the appropriate setting of the MXT value is to try a range of values during peak system times while monitoring the system's use of the processor and storage. There are a few comments about applying this technique:

- Note that after CICS Version 1.7, a Dispatch Control Area (DCA) is built for any task received from VTAM regardless of whether the MXT (or CMXT) ceiling has been reached.

- A Task Control Area (TCA) is created at dispatch time (which is where the decision is made as to whether or not the region is at MXT or CMXT). If the region is at MXT or CMXT at dispatch time, the TCA is not created.

- Therefore, the MXT limit in CICS 1.7 is a limit on the number of tasks that are permitted to acquire a TCA. A task is not included in the AMXT calculation until it has been permitted to acquire its TCA even though its DCA appears on the chain of "active" tasks.

- CEMT looks at the DCA chain and reports on the existence of DCAs. This reported value is very misleading as to the number of tasks in the system.

CPExpert suggests that you not simply set MXT to its maximum value (999). Such a value can seriously impact performance in CICS Version 3.1.1 and later versions. See Rule CIC104 for additional information.

If you are an MRO environment, be sure to set MXT greater than the number of sessions allowed. (Alternatively, lower the number of allowed sessions to less than the MXT value - whichever best fits your environment.)

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 48 and 192.

CICS/MVS Version 2.1.2 Performance Guide: pages 235 and 376.

CICS/ESA Version 3.1.1 Performance Guide: page 53, page 295, and page 307.

CICS/ESA Version 3.2.1 Performance Guide: page 191, page 203, and page 271.

CICS/ESA Version 3.3.1 Performance Guide: page 201, page 221, and page 291.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.7.3, Appendix A.1.4, and Appendix C.7.

CICS/TS Release 1.1 Performance Guide: Section 4.7.3, Appendix 1.1.3, and Appendix 1.4.8.

CICS/TS Release 1.2 Performance Guide: Section 4.7.4, Appendix 1.1.4. and Appendix 1.1.29.

CICS/TS Release 1.3 Performance Guide: Section 4.11.3, Appendix 1.1.5. and Appendix 1.1.32.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (MXT) and Appendix A (Table 131).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.3 Setting the maximum task specification (MXT)

CICS MRO Tuning and Performance Guide: page 37.

IBMLINK, Document Q579023.

IBMLINK, Document Q465409.

Rule CIC102: The AMXT value may be too low

Finding: CPExpert has detected that the CICS region reached the active maximum tasks (AMXT value) and virtual storage was not a constraint.

Impact: This finding has a MEDIUM IMPACT on the performance of the CICS region. This rule does not apply with CICS Version 4.1.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The AMXT operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks in the CICS region. All new and resumed tasks must pass the AMXT limit before CICS dispatches the tasks. CICS marks all new and resumed tasks "nondispatchable for AMXT reasons."

Each message entering the system causes a Dispatch Control Area (DCA) to be created (this logic is for CICS Version 1.7 and later versions). It the MXT value has not been reached, CICS creates a Task Control Area (TCA). After the TCA has been created, CICS considers whether the task should be dispatched. It is at this point that the AMXT value is used; tasks are selected for dispatching only if fewer than AMXT tasks are already active (that is, they are on the active chain). (Note that journal control tasks and the terminal control task are dispatched without regard to the AMXT value.)

The AMXT value determines how far down the active chain of TCAs CICS task control will search, looking for a dispatchable task, before issuing an operating system WAIT. The AMXT operand does not control the number of tasks allowed on the active chain.

As CICS is scanning the active chain it will take action based on the code in TCATCDC. Exhibit CIC102-1[1] illustrates the actions based on the state of the task.

---

[1]Source: IBMLINK Document ID Q361783
TITLE: HOW IS AMXT VALUE AFFECTED BY TASKS WAITING FOR JOURNAL OR FILE CONTROL?

TCATCDC	AMXT	IOCP	Dispatch	ABEND	DUMP	State
10	N	N	No	No	No	Non-Dispatchable
14	Y	N	Not AMXT	No	No	AMXT Reason Code
15	N	N	Not CMXT	No	No	CMXT Reason Code
17	N	N	PCA available	No	No	Waiting for PCA
20	N	N	Yes	No	No	Dispatchable
21	N	N	Yes	Yes	Yes	ABEND
22	N	N	Yes	Yes	No	Stall Purge
24	N	N	Yes	Yes	No	RTIMOUT Expired
25	N	N	Yes	Yes	No	DTIMOUT Expired
40	N	N	If Posted	No	No	WAIT on List
41	Y	N	If Posted	No	No	Page I/O WAIT
43	Y	Y	If Posted	No	No	I/O Event WAIT
80	Y	N	If Posted	No	No	Single Event WAIT
88	N	N	If Posted	No	No	CICS Event WAIT

CICS TASK DISPATCHER ACTIONS ON ACTIVE CHAIN

EXHIBIT CIC102-1

Tasks on the active chain waiting for I/O events, Page I/O and single event WAITs are counted for AMXT. Once the AMXT value is reached, no more tasks are eligible to be dispatched (unless they are system tasks). If no tasks are dispatched for the stall interval (ICVS), AMXT is temporarily set to 9999 and the scan is repeated to prevent a deadlock.

There are several situation in which the AMXT value may be used to limit the number of active tasks.

- Limiting the number of active tasks may be used to limit the central storage required to support the working set of the active tasks. If central storage is a system constraint, management may wish to limit the amount of central storage required to support CICS tasks. The AMXT value can be used to limit the number of active tasks and, consequently, limit the central storage required to support their working sets.

- Management may wish to limit the effect that a CICS region has on the rest of the system or on other CICS regions. For example, management may wish to limit the number of active tasks in a CICS test region. By limiting the number of active tasks in the test region, management can control the resources consumed by CICS testing.

- AMXT can be used to control the use of virtual storage by the Dynamic Storage Area (DSA). When the number of dispatchable tasks reaches the AMXT value, tasks on the active chain are not dispatched. Consequently, the tasks generally have little associated storage, or their storage is likely to be paged out of central storage. The AMXT value can be particularly useful if there are a number of long-running tasks and the MXT value cannot be successfully used to limit the amount of virtual storage used by the tasks.

There are a number of disadvantages to limiting the maximum number of active tasks.

- CICS performance can be unnecessarily degraded if the AMXT value is used to restrict the number of concurrent active tasks, and if the system is capable of handling more tasks.

- Tasks are selected from the dispatchable chain based upon priority. Active tasks may not be dispatched because they are waiting for some event, such as waiting on I/O completion. If the AMXT value is too low, it is possible that no tasks are dispatched even though low-priority dispatchable tasks may be on the active chain.

- A temporary lockout can occur if the AMXT value is set too low in an environment where tasks are dependent upon the completion of events processed by other tasks. For example, one task might attach another task and then wait for the completion of an event processed by the attached task. This situation is known as a "maximum active stall condition" and may seriously degrade performance.

  (The situation can be prevented by specifying a sufficiently high AMXT value. Alternatively, the situation can be prevented by assigning tasks that wait on events from other tasks to a task class and specifying a CMXT value that is lower than the AMXT value.)

The CICS statistics report only the number of active tasks reached. The statistics do not report how many times the AMXT value limited the number of active tasks. CPExpert produces Rule CIC102 if the CICS statistics reported that number of maximum active tasks was as large as the AMXT value, and CPExpert did not detect a virtual storage constraint. **Note that a central storage constraint may have existed but would have been undetectable from the CICS statistics.**

You may have set the AMXT value to specifically limit the number of tasks, and may not wish CPExpert to provide notification that the AMXT value was reached. This rule is applied only if **%LET AMAXTASK = Y;** is specified in USOURCE(CICGUIDE). You can "turn off" this rule by specifying **%LET AMAXTASK = N;** in USOURCE(CICGUIDE).

Suggestion: CPExpert suggests that you consider increasing the AMXT value in the SIT. An effective way of finding the appropriate setting of the AMXT value is to try a range of values during peak system times while monitoring the system's use of the processor and storage.

Note that the AMAX value may have been set to restrict the amount of central storage used by CICS transactions. There is no information in the CICS statistics to indicate whether central storage is a constraint. If central storage is a constraint to overall system performance, the above suggestion to increase the AMXT value should be ignored.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 48 and 195.

CICS/MVS Version 2.1.2 Performance Guide: pages 238 and 376.

CICS/ESA Version 3.1.1 Performance Guide: pages 53 and 296.

CICS/ESA Version 3.2.1 Performance Guide: pages 192 and 271.

CICS/ESA Version 3.3.1 Performance Guide: pages 202 and 291.

CICS/ESA Version 4.1.1: not applicable

CICS Transaction Server for OS/390: not applicable

CICS/TS for z/OS: not applicable

IBMLINK, Document Q361783.

Rule CIC103: Maximum Task Class was reached too often

Finding: CPExpert has detected that the CICS region reached the maximum task class specification (CMXT value) too often.

Impact: This finding has a MEDIUM IMPACT on the performance of the CICS region. This rule does not apply with CICS Version 4.1.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The Maximum Task Class (CMXT) operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks associated with particular task classes. Up to 10 unique task classes can be defined, and a maximum number of active tasks is specified for each class.

Tasks are assigned to classes using the TCLASS operand in the Program Control Table (PCT). For example, CMXT might be defined as "CMXT=(10,4,4,20)" in the SIT. Four task classes are thus defined. Class 1 has a maximum of 10 active tasks, Class 2 and Class 3 have a maximum of 4 active tasks, and Class 4 has a maximum of 20 active tasks.

Note that after CICS Version 1.7, a Dispatch Control Area (DCA) is built for any task received from VTAM regardless of whether the MXT or CMXT ceiling has been reached. The CMXT values do **not** limit the number of tasks which are ATTACHed. Rather, the CMXT values are applied to the task classes when a task is to be considered for dispatching.

After the ATTACH has been processed, CICS considers whether the task should be dispatched. A Task Control Area (TCA) is created at dispatch time if the region is below MXT or if the task class is below CMXT. It is at this point that the CMXT value is used.

Tasks assigned to a task class are selected for dispatching only if fewer than CMXT tasks are already active for the class. If the region is at the MXT or CMXT limit at dispatch time, the TCA is not created.

There are several situation in which the CMXT value may be used to limit the number of active tasks in a class.

- The CMXT specifications can be used to control tasks that may be heavy resource users. This control can limit the amount of resources required to support the heavy resource users (either because management wishes to restrict the amount of resources allowed to

CICS or because management wishes to restrict the resources devoted to specific tasks).

- The CMXT specifications can be used to control low priority tasks. This control can allow processor resources to be used by more important tasks.

- The CMXT operand can be used to ensure that one type of transaction does not monopolize CICS.

- CMXT can be used to control the use of virtual storage by the Dynamic Storage Area (DSA). When the number of dispatchable tasks reaches the CMXT value, tasks on the active chain are not dispatched. Consequently, the tasks generally have little associated storage, or their storage is likely to be paged out of central storage. The CMXT value can be particularly useful if there are a number of long-running tasks and the MXT value cannot be successfully used to limit the amount of virtual storage used by the tasks.

  The CMXT operand normally should not be used to serialize tasks. Rather, ENQ should be used for this purpose.

There are a number of disadvantages to limiting the maximum number of active tasks.

- CICS performance can be unnecessarily degraded if the CMXT value is used to restrict the number of concurrent active tasks in a particular class, and if the system is capable of handling more tasks.

- CMXT is not normally suited for conversational transactions, because users can be locked out for a lengthy period if the CMXT value is reached.

- Tasks assigned to a particular class are selected from the dispatchable chain based upon priority. Active tasks may not be dispatched because they are waiting for some event, such as waiting on I/O completion. If the CMXT value is too low, it is possible that no tasks are dispatched even though low-priority dispatchable tasks may be on the active chain. (Recall that the AMXT operand controls how far down the dispatch chain CICS task control searches looking for a dispatchable task. It **does not** control the number of dispatched tasks, but controls the number of **potentially** dispatchable tasks.)

- A temporary lockout can occur if the CMXT value is set too low in an environment where tasks are dependent upon the completion of events processed by other tasks. For example, one task might attach another task and then wait for the completion of an event processed by the attached task. If the attached task is assigned to a task class and the task class is at its maximum, considerable delay may result.

CPExpert produces Rule CIC103 if the CICS stastics reported that the maximum active tasks value for any class was reached more than the CMAXn guidance value for the class, and if the maximum number of concurrently attached tasks for the task class was greater than the CMXT specification in the SIT for the task class.

Suggestion: CPExpert suggests that you review the CMXT value in the SIT. Under most circumstances, the CMXT ceiling should be reached infrequently. If the ceiling is frequently reached, you may wish to increase the ceiling.

Alternatively, reaching the CMXT ceiling may simply be an indication of a performance problem elsewhere in CICS. If CICS performance is poor, tasks will not quickly flow through the system and the CMXT ceiling may be reached. Tuning CICS areas may cause tasks to move through the system quicker and the CMXT ceiling may not restrict the number of tasks dispatched in the class.

Alternatively, you may wish to direct tuning efforts at the specific tasks assigned to the class. Perhaps the resource demands of the tasks can be reduced.

Alternatively, the resources used by the tasks can be a constraint. For example, DASD constraints may impede the flow of tasks assigned to the class. Perhaps a improvement in the DASD configuration, reorganizing file placement on volumes, or other actions would reduce the DASD constraint and allow the tasks to more quickly flow through the system.

.

If the CMXT value is appropriate for your management objectives and CICS performance cannot be improved, then you may wish to change the CMAXn guidance value for the class to prevent spurious production of this rule.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 48 and 199.

CICS/MVS Version 2.1.2 Performance Guide: pages 242 and 377.

CICS/ESA Version 3.1.1 Performance Guide: pages 123 and 298.

CICS/ESA Version 3.2.1 Performance Guide: pages 194 and 337.

CICS/ESA Version 3.3.1 Performance Guide: pages 204 and 355.

CICS/ESA Version 4.1.1: not applicable.

CICS Transaction Server for OS/390: not applicable

CICS/TS for z/OS: not applicable

Rule CIC104: Maximum Task (MXT) specification may be too large

Finding: CPExpert has detected that the value specified for the MXT operand in the System Initialization Table (SIT) may be too large.

Impact: This finding has a MEDIUM IMPACT on the performance of the CICS region. This finding applies only to CICS/ESA Version 3.1.1 and later versions.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Please refer to Rule CIC101 for additional information about the MXT operand.

Some installations have specified MXT=999, with the intention of allowing an unlimited number of tasks in storage. That is, the number of tasks will be limited only by the demands of CICS users. Additionally, specifying MXT=999 is recommended by IBM's CICS MOR Tuning and Performance Guide.

Specifying a large value for MXT had little adverse affect before CICS/ESA Version 3.1.1. However, for CICS/ESA Version 3.1.1 and later versions, this specification can have an adverse effect on available storage and on CICS performance.

With CICS/ESA Version 3.1.1 and later versions, CICS uses the MXT parameter to determine the kernel storage allocation. This allocation is calculated in 4-K, 5K, and 8K byte units. (The number of units is derived based upon the value of the MXT operand.) The CICS Performance Guide explains that the kernel stack storage allocations are made as follows:

- Below the 16-megabyte line:
  - One 5K-byte segment for each task, up to the MXT limit, plus four, **and**
  - One 4K-byte segment for every 10 tasks (based on MXT, plus four, rounded down).
- Above the 16-megabyte line:

- One 8K-byte segment for each task, up to the MXT limit, plus four, **and**

- One 4K-byte segment for every 10 tasks (based on MXT, plus four, rounded down).

Thus, if you specify MXT=100 in the SIT, CICS requires the following storage:

- Below the line: (104 * 5K) + (10 * 4K) = 560K bytes

- Above the line: (104 * 8K) + (10 * 4K) = 872K bytes

Specifying MXT=100 might be a reasonable specification in many environments and the above calculation shows the storage required to support these tasks. However, if you specify MXT=999, the amount of storage is increased significantly!

- Below the line: (1003 * 5K) + (100 * 4K) = 5,415K bytes

- Above the line: (1003 * 8K) + (100 * 4K) = 8,424K bytes

Few installations would require such a high specification for MXT, and most installations would begin to experience CICS stress and short-on-storage conditions because so much storage was required for the kernel stacks. The space allocated for kernel stacks is **permanently** allocated - so MXT=999 cannot be specified just for initialization with the intent of lowering the value during operation. The storage will be allocated based upon the value CICS is provided during initialization, and remains permanently allocated.

If the maximum number of tasks specified on the MXT operand is too high for the available storage, CICS will not be able to obtain the amount of MVS storage it needs for kernel stack segments. In this case, CICS progressively reduces the MXT value until it is able to obtain the MVS storage that it needs. CICS reduces the MXT value using the following algorithm:

$$MXT = \frac{MXT - 32}{2} + 32$$

Having a large value for a Terminal Owning Region (TOR) is not usually detrimental unless the region is also an AOR region. TOR regions use their space simply for queuing, and thus generally do not experience short-on-storage conditions.

CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is less than the **PCTMXTLO** guidance variable, CPExpert concludes that the MXT value may be too high. Rule CIC104 is produced to report this conclusion.

If the condition exists for **any** CICS statistics interval in the data being analyzed for a region, CPExpert lists the percent of active tasks as a function of all intervals, and places '***' beside those intervals in which the conclusion was reached. This listing is done to give a sense of perspective of (1) how often and (2) when the condition was detected.

Initially, CPExpert arbitrarily produced Rule CIC104 if the number of active tasks was less than 75% of the MXT value. The **PCTMXTLO** value was introduced to allow users to specify guidance to CPExpert.

Suggestion: CPExpert suggests that you review the MXT value in the SIT and determine why it is specified so much higher than required by normal CICS operation. Unless there are unusual circumstances, you may wish to specify the MXT operand with a value more representative of the tasks actually processed by the region.

The MXT ceiling normally should not be reached unless management wishes to restrict access to the CICS region.

The value for the MXT operand normally should be sufficiently high that tasks are not restricted by MXT. Of course, if storage is a constraint, you might have to use the MXT operand to restrict access to CICS by tasks. However, be very careful not to allow the MXT value to be reached in a MRO environment (unless you have a very unusual environment).

Alternatively, you can adjust CPExpert's analysis by using the PCTMXTLO guidance to specify different guidance to CPExpert.

This rule probably should be ignored if the region is a TOR-only region and you are executing in Compatibility Mode. There usually is little harm done by allocating more kernel stacks than required in a TOR region for Compatibility Mode.

The rule **does** apply for a TOR if you are executing in Goal Mode, since unnecessary overhead would be generated by MVS in scanning Performance Blocks created for each possible task.

Reference: CICS/OS/VS Version 1.7 - Not applicable.

CICS/MVS Version 2.1.2 - Not applicable.

CICS/ESA Version 3.1.1 Performance Guide: page 53, page 295, and page 307.

CICS/ESA Version 3.2.1 Performance Guide: page 191, page 203, and page 271.

CICS/ESA Version 3.3.1 Performance Guide: page 201, page 221, and page 291.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.7.3, Appendix A.1.4, and Appendix C.7

CICS/TS Release 1.1 Performance Guide: Section 4.7.3, Appendix 1.1.28, and Appendix 4.8.

CICS/TS Release 1.2 Performance Guide: Section 4.7.4, Appendix 1.1.4. and Appendix 1.1.29.

CICS/TS Release 1.3 Performance Guide: Section 4.11.3, Appendix 1.1.5. and Appendix 1.1.32.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (MXT) and Appendix A (Table 131).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.3 Setting the maximum task specification (MXT)

CICS MRO Tuning and Performance Guide: page 37.

IBMLINK, Document Q579023.

IBMLINK, Document Q465409.

Rule CIC105: Transaction class reached MAXACTIVE too often

Finding: CPExpert has detected that too many tasks in a transaction class were queued because the MAXACTIVE limit had been reached.

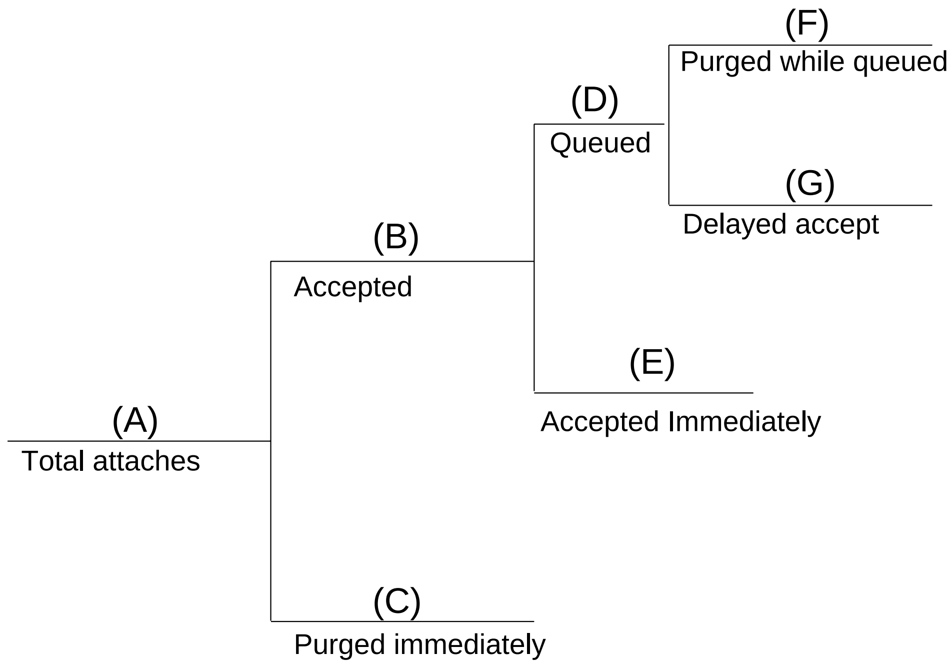
Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The MAXACTIVE attribute for a transaction class can be used to control the number of active tasks in the transaction class. The MAXACTIVE attribute limits the number of transactions for a specific transaction class, while the MXT value (specified in the System Initialization Table) limits the total number of transactions in the CICS region.

The MAXACTIVE value is specified for a transaction class via the CEDA transaction definition with the TRANCLASS keyword. The MAXACTIVE value can be changed using the CEMT SET TRANCLASS command or using the EXEC CICS SET TRANCLASS command.

Please refer to CIC105-1 for a graphical representation of the processing of tasks.



Source: IBM *CICS Performance Guide, CICS Version 4.1*

TRANSACTION CLASS PROCESSING

EXHIBIT CIC105-1

- Attach requests are processed by the CICS region for tasks assigned to a transaction class, shown as "(A)" in Exhibit 105-1. This value is contained in the XMCTAT variable of the transaction class statistics.

- Attach requests are either (1) purged immediately or (2) processed further.

The attach requests will be purged immediately if the number of tasks currently accepted or queued equals the PURGETHRESH attribute of the transaction class definition. The number of attach requests that were purged immediately (shown as "(C)" in Exhibit CIC105-1) is contained in the XMCPI variable of the transaction class statistics.

If not purged immediately, the attach requests will be processed further.

- Attach requests which are not purged immediately may be accepted immediately or may be queued.

The attach requests are accepted immediately if the number of currently active tasks is less than the MAXACTIVE attribute of the transaction class definition. The number of attach requests that were accepted immediately (shown as "(E)" in Exhibit CIC105-1) is contained in the XMCAI variable of the transaction class statistics.

An attach request is queued if the number of currently active tasks is equal to or greater than the MAXACTIVE attribute of the transaction class definition. The number of attach requests that were queued (shown as "(D)" in Exhibit CIC105-1) is contained in the XMCAAQ and XMCPWQ variables of the transaction class statistics (the two variables must be summed to yield the total number of attach requests that were queued).

- An attach request that is queued may be purged by operator action while it is queued. The number of attach requests that were purged while queued (shown as "(F)" in Exhibit CIC105-1) is contained in the XMCPWQ variable of the transaction class statistics.

- If the attach request is not purged by operator action, the request is accepted after some queuing delay. The number of attach requests that were accepted after being queued (shown as "(G)" in Exhibit CIC105-1) is contained in the XMCAAQ variable of the transaction class statistics.

As described above, the MAXACTIVE attribute of the transaction class definition is used to control the number of tasks which may be active in the CICS region at any one time. There are several situation in which the MAXACTIVE attribute may be used to limit the number of active tasks in a class.

- The MAXACTIVE attribute can be used to control tasks that may be heavy resource users. This control can limit the amount of resources required to support the heavy resource users (either because management wishes to restrict the amount of resources allowed to CICS or because management wishes to restrict the resources devoted to specific tasks).

- The MAXACTIVE attribute can be used to control low priority tasks. This control can allow processor resources to be used by more important tasks.

- The MAXACTIVE operand can be used to ensure that one type of transaction does not monopolize CICS.

- **The MAXACTIVE operand normally should not be used to serialize tasks. Rather, ENQ should be used for this purpose**.

There are a number of disadvantages to limiting the maximum number of active tasks.

- CICS performance can be unnecessarily degraded if the MAXACTIVE value is used to restrict the number of concurrent active tasks in a particular class, and if the system is capable of handling more tasks.

- MAXACTIVE is not normally suited for conversational transactions, because users can be locked out for a lengthy period if the MAXACTIVE value is reached.

- Tasks assigned to a particular class are selected from the dispatchable chain based upon priority. Active tasks may not be dispatched because they are waiting for some event, such as waiting on I/O completion. If the MAXACTIVE value is too low, it is possible that no tasks are dispatched even though low-priority dispatchable tasks may be on the active chain.

- A temporary lockout can occur if the MAXACTIVE value is set too low in an environment where tasks are dependent upon the completion of events processed by other tasks. For example, one task might attach another task and then wait for the completion of an event processed by the attached task. If the attached task is assigned to a task class and the task class is at its maximum, considerable delay may result.

IBM suggests that transaction classes be established for the following categories of transactions:

- Simple enquiries
- Complex enquiries or short browses
- Long browses
- Short updates
- Long updates

CPExpert produces Rule CIC105 if the CICS interval statistics reported that the maximum active tasks value for any class was reached more than the corresponding guidance value for the transaction class.

Suggestion: CPExpert suggests that you review the MAXACTIVE attribute for the transaction class. Under most circumstances, the MAXACTIVE ceiling should be reached infrequently. If the ceiling is frequently reached, you may wish to increase the ceiling.

Alternatively, reaching the MAXACTIVE ceiling may simply be an indication of a performance problem elsewhere in CICS. If CICS performance is poor, tasks will not quickly flow through the system and the MAXACTIVE ceiling may be reached. Tuning CICS areas may cause tasks to move through the system quicker and the MAXACTIVE ceiling may not restrict the number of tasks dispatched in the class.

Alternatively, you may wish to direct tuning efforts at the specific tasks assigned to the class. Perhaps the resource demands of the tasks can be reduced.

Alternatively, the resources used by the tasks can be a constraint. For example, DASD constraints may impede the flow of tasks assigned to the class. Perhaps a improvement in the DASD configuration, reorganizing file placement on volumes, or other actions would reduce the DASD constraint and allow the tasks to more quickly flow through the system.

If the MAXACTIVE attribute for the transaction class identified by this rule is appropriate for your management objectives and CICS performance cannot be improved, then you may wish to change the guidance value for the transaction class to prevent the spurious firing of this rule.

Reference: *CICS/ESA Version 4.1 Performance* Guide: Section 4.7.4 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 4.7.5 and Appendix 1.1.27.

CICS/TS Release 1.2 Performance Guide: Section 4.7.5 and Appendix 1.1.28.

CICS/TS Release 1.3 Performance Guide: Section 4.11.4 and Appendix 1.1.31.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (MAXACTIVE) and Appendix A (Table 129)

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.4 Using transaction classes (MAXACTIVE) to control transactions

Rule CIC106: Too many tasks were queued for MAXACTIVE reason

Finding: CPExpert has detected that too many tasks in a transaction class were queued because the MAXACTIVE limit had been reached, the PURGETHRESH value had not limited the number of queued tasks, and the CICS region encountered a Short-on-Storage condition.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: The following rule caused this rule to be invoked:

CIC110: CICS encountered a Short-on-Storage condition

Discussion: Please refer to Rule CIC105 for a discussion of the processing of transaction tasks: how the transactions are accepted, purged, or queued for processing.

When the number of active transactions reaches the MAXACTIVE value, attach requests are queued.

Each queued transaction requires only 256 bytes plus the size of the terminal input/output area (TIOA) holding any terminal input. However, if a large number of transactions are queued, the queue could grow to occupy all available storage in the CICS Dynamic Storage Area (DSA). CICS then could become short-on-storage, with serious performance consequences.

The PURGETHRESH value controls the number of newly-created tasks that are accepted but must be queued because of MAXACTIVE reasons. A limit on the number of queued transactions is necessary in some situations because a very large number of transactions might be queued in a heavily-loaded system.

CPExpert produces Rule CIC106 if the CICS interval statistics reported that the CICS region encountered a short-on-storage situation, and if transactions were queued for MAXACTIVE reason.

Rule CIC106 is the counterpart of Rule CIC107: Rule CIC107 suggests that the PURGETHRESH value be increased because transactions were purged but no short-on-storage situation occurred; Rule CIC106 notes that a short-on-storage situation was encountered, but transactions were queued.

CPExpert estimates and reports the storage used by the queued transactions by multiplying the peak number of queued transactions by 256 bytes. This calculation results in a low value, as the TIOA is not included.

Suggestion: CPExpert suggests that you review the PURGETHRESH attribute for the transaction class. Under most circumstances, the PURGETHRESH ceiling should be high enough that transactions are queued but are not purged.

However, when the CICS region encounters a short-on-storage situation, the PURGETHRESH is one way in which storage may be regained. You can reduce the number of queued transactions by specifying a lower value for the PURGETHRESH attribute for the transaction class.

CPExpert suggests that you first evaluate other alternatives to lowering the PURGETHRESH value. The PURGETHRESH attribute is intended to prevent a significant amount of storage being used by queued transactions. This situation might occur when the CICS region slows down or when an unexpectedly large number of transactions attempt to attach.

Before specifying a smaller value for the PURGETHRESH attribute, you should explore other alternatives. Some alternatives are:

- Improve the overall performance of the CICS region so transactions are processed faster. This might involve adjusting other CICS parameters, giving CICS a higher dispatching priority, rescheduling work so that CICS does not compete for resources, acquiring additional resources, improving overall system performance, etc.

- Impose MAXACTIVE limits on transaction classes which are heavy resource users. This option would be particularly applicable if you can identify high resource using transactions, place these transactions into their own transaction class, and impose a MAXACTIVE limit on the high resource using transaction class. This alternative might allow the CICS region to process the transactions which use few resources.

- Implement task prioritization, and give a low priority to tasks which are use a large amount of system resources. Note that CICS does not implement a preemptive dispatching algorithm. This alternative might not have any positive effect to reduce the queue of transactions if the tasks that use a large amount of CPU resources are dispatched and they fail to release the processor.

There are, of course, other alternatives you should consider before adjusting the PURGETHRESH value.

Reference: *CICS/ESA Version 4.1 Performance* Guide: Section 4.7.5 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 4.7.5 and Appendix 1.1.27.

CICS/TS Release 1.2 Performance Guide: Section 4.7.5 and Appendix 1.1.28.

CICS/TS Release 1.3 Performance Guide: Section 4.11.4 and Appendix 1.1.31.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (MAXACTIVE) and Appendix A (Table 129).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.4 Using transaction classes (MAXACTIVE) to control transactions |

Rule CIC107: PURGETHRESH value should be increased

Finding: CPExpert has detected that the PURGETHRESH value limited the number of queued tasks but the CICS region had not encountered a Short-on-Storage condition.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: The following rules cause this rule to be suppressed:

- CIC106: Too many tasks were queued for MAXACTIVE reason
- CIC110: CICS encountered a Short-on-Storage condition

Discussion: Please refer to Rule CIC105 for a discussion of the processing of transaction tasks: how the transactions are accepted, purged, or queued for processing.

When the number of active transactions reaches the MAXACTIVE value, attach requests are queued.

Each queued transaction requires only 256 bytes plus the size of the terminal input/output area (TIOA) holding any terminal input. However, if a large number of transactions are queued, the queue could grow to occupy all available storage in the CICS Dynamic Storage Area (DSA). CICS then could become short-on-storage, with serious performance consequences.

The PURGETHRESH value controls the number of newly-created tasks that are accepted but must be queued because of MAXACTIVE reasons. A limit on the number of queued transactions is necessary in some situations because a very large number of transactions might be queued in a heavily-loaded system.

CPExpert produces Rule CIC107 if the CICS interval statistics reported that transactions were purged because the PURGETHRESH value had been reached, but the CICS region had not encountered a short-on-storage situation.

Rule CIC107 is the counterpart of Rule CIC106: Rule CIC106 suggests that the PURGETHRESH value be reduced because a short-on-storage situation occurred; Rule CIC107 notes that no short-on-storage situation was encountered, but transactions were purged. CPExpert suppresses Rule CIC107 if Rule CIC106 is produced during any analysis interval.

CPExpert estimates and reports the storage used by the queued transactions by multiplying the peak number of queued transactions by 256 bytes. This calculation results in a low value, as the TIOA is not included.

Suggestion: CPExpert suggests that you review the PURGETHRESH attribute for the transaction class. Under most circumstances, the PURGETHRESH ceiling should be high enough that transactions are queued but are not purged.

However, when the CICS region encounters a short-on-storage situation, the PURGETHRESH is one way in which storage may be regained. The PURGETHRESH attribute is intended to prevent a significant amount of storage being used by queued transactions. This situation might occur when the CICS region slows down or when an unexpectedly large number of transactions attempt to attach.

Before taking any action to increase the PURGETHRESH, you should wait for a reasonable interval (perhaps a week or so) to ensure that Rule CIC106 is not produced.

Before specifying a smaller value for the PURGETHRESH attribute, you should explore other alternatives (improve the overall performance of the CICS region so transactions are processed faster, impose MAXACTIVE limits on transaction classes which are heavy resource users, etc.).

Reference: *CICS/ESA Version 4.1 Performance* Guide: Section 4.7.5 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 4.7.5 and Appendix 1.1.27.

CICS/TS Release 1.2 Performance Guide: Section 4.7.6 and Appendix 1.1.28.

CICS/TS Release 1.3 Performance Guide: Section 4.11.5 and Appendix 1.1.31.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (PURGETHRESH) and Appendix A (Table 129)

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.5 |
Specifying a transaction class purge threshold (PURGETHRESH) |

Rule CIC108: Maximum Task (MXT) specification may be too small

Finding: CPExpert has detected that the value specified for the MXT operand in the System Initialization Table (SIT) may be too small.

Impact: This finding has a NO IMPACT on the performance of the CICS region, but it is an "early warning" that a problem might occur.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The MXT operand in the System Initialization Table (SIT) limits the total number of concurrent tasks in the CICS region. Please refer to Rule CIC101 for additional information about the MXT operand.

Specifying a "correct" value for the MXT keyword is a balance between (1) specifying a value that is too high and (2) specifying a value that is too low.

If the MXT value is too high, storage is wasted and (with Goal Mode) unnecessary overhead is generated. The logic associated with Rule CIC104 deals with situation in which the MXT value is too high.

If the MXT value is too low, CICS will fail to attach tasks when the number of tasks reaches the MXT value. Rule CIC101 provides an indication of the number of times that the number of active tasks reached the MXT value.

This indication that MXT was reached is acceptable for many CICS regions. However, reaching MXT might be unacceptable for some critical CICS regions. Installation personnel need to be aware of the possibility of a too-low MXT value for critical CICS regions. The logic associated with Rule CIC108 can be used to provide an alert that MXT value might be too low.

CPExpert computes the percent of active tasks as a function of the value of the MXT keyword. When the percent of active tasks is greater than the **PCTMXTHI** guidance variable, CPExpert concludes that the MXT value may be too low. Rule CIC108 is produced to report this conclusion.

The default value for the PCTMXTHI guidance variable is 100%, which effectively "turns off" the logic in Rule CIC108 (the percent cannot be higher than 100%). You can specify an appropriate percentage if you have critical CICS regions and wish to be alerted that the number of active tasks may be in danger of reaching the MXT specification. CPExpert will produce Rule CIC108 when the number of active tasks reaches the specified percent of MXT.

Suggestion: CPExpert suggests that you review the information provided by Rule CIC108 and determine whether the number of active tasks is likely to reach the MXT value specified in the SIT.

The value for the MXT operand normally should be sufficiently high that tasks are not restricted by MXT. This particularly is true for important CICS regions, unless you have deliberately specified a lower MXT value for storage constraint considerations.

Reference: CICS/OS/VS Version 1.7 - Not applicable.

CICS/MVS Version 2.1.2 - Not applicable.

CICS/ESA Version 3.1.1 Performance Guide: page 53, page 295, and page 307.

CICS/ESA Version 3.2.1 Performance Guide: page 191, page 203, and page 271.

CICS/ESA Version 3.3.1 Performance Guide: page 201, page 221, and page 291.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.7.3, Appendix A.1.4, and Appendix C.7

CICS/TS Release 1.1 Performance Guide: Section 4.7.3, Appendix 1.1.28, and Appendix 4.8.

CICS/TS Release 1.2 Performance Guide: Section 4.7.4, Appendix 1.1.4. and Appendix 1.1.29.

CICS/TS Release 1.3 Performance Guide: Section 4.11.3, Appendix 1.1.5. and Appendix 1.1.32.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (MXT) and Appendix A (Table 131).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.3 Setting the maximum task specification (MXT) |

Rule CIC109: CICS region is approaching maximum capacity

Finding: CPExpert has detected that the amount of CPU used by the quasi-reentrant (QR) mode TCB is approaching the CPU capacity of the region.

Impact: This finding has no direct impact on the performance of the CICS REGION. However, the finding provides an "early warning" that the region is approaching maximum capacity.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS always has two or three TCBs for normal processing, depending on the release of CICS:

- The quasi-reentrant (**QR** mode) TCB executes the quasi-reentrant application code and most CICS code. This TCB is available in all releases of CICS.
- The resource-owning (**RO** mode) TCB is used for opening and closing data sets and for program loading. This TCB is available in all releases of CICS.
- The file-owning (**FO** mode) TCB is used for opening and closing data sets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 1.

CICS optionally has other TCBs, depending on the release of CICS.

If the Monitoring Class option is ON (MNPER="ON" in the System Initialization Table), the dispatcher domain maintains an "accumulated CPU TCB time" in the DSGACT variable.

Additionally, the dispatcher maintains "accumulated time dispatched" and "accumulated time in MVS wait" variables (DSGTDT and DSGTWT, respectively). The sum of the *accumulated time dispatched* and *accumulated time in MVS wait* is approximately the elapsed time that CICS was operational.

Dividing the *accumulated CPU TCB time* by the sum of the *accumulated time dispatched* and *accumulated time in MVS wait* yields an approximation of the **percent CPU busy** of the CICS region.

The DSGACT value does not include uncaptured CPU time, so the value normally will be less than the amount of CPU time actually used by the CICS region. IBM's *CICS Performance Guides* state that even with a totally busy CICS region, the calculated percent CPU busy of the CICS region would not normally be 100%. Consequently, IBM suggests that a region should be considered approaching maximum capacity if the calculated percent CPU busy exceeds 70%.

CPExpert computes the percent CPU Busy by the following algorithm:

$$Percent\ CPU\ Busy\ for\ CICS\ region = \frac{DSGACT}{DSGTDT + DSGTWT}$$

where:

DSGACT = Accumulated CPU TCB time
DSGTDT = Accumulated time dispatched
DSGTWT = Accumulated time in MVS wait

CPExpert produces Rule CIC109 when the Percent CPU Busy is greater than the value specified by the **PCTQRTCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTQRTCB** guidance variable is 60% indicating that CPExpert should produce Rule CIC109 whenever more than the Percent CPU Busy was more than 60% for the CICS region. The default guidance value is less than the 70% suggested by IBM to give an "early warning" of a capacity restraint.

Suggestion: CPExpert suggests that you review the information provided by Rule CIC109 and determine whether the Percent CPU Busy is reasonable, and is stable. If the Percent CPU Busy is increasing over time, you should consider the following alternatives:

- Review the applications to determine whether the volume of data is increasing and whether the increase is reasonable.

- Review the applications to determine whether coding errors (such as looping) or inefficiencies in design cause increased CPU usage.

- If you have more than one CPU, consider splitting the CICS region and allocating the workload between the resulting CICS regions.

- You can modify the **PCTQRTCB** guidance variable to control the percent CPU busy at which Rule CIC109 is produced.

Reference: *CICS/ESA Version 4.1.1 Performance* Guide: Section 2.2.7.1 and Appendix A.1.4.

CICS/TS Release 1.1 Performance Guide: Section 2.2.7.1 and Appendix 1.1.3.

CICS/TS Release 1.2 Performance Guide: Section 2.2.8.1 and Appendix 1.1.4.

CICS/TS Release 1.3 Performance Guide: Section 2.2.8.1 and Appendix 1.1.5.

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.7 and Appendix 1.5.

Thanks: Thanks to Rexaldo Avendano (Kaiser Permanente) for suggesting this rule.

Rule CIC110: CICS encountered a Short-on-Storage condition

Finding: CPExpert has detected that the CICS region encountered a short-on-storage condition.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS tries to keep a minimum amount of free space in the dynamic storage area (DSA) to respond to GETMAIN requests from tasks. This is accomplished by performing "program compression" whenever the number of available pages in the DSA falls below a threshold or whenever a GETMAIN cannot be satisfied from available free storage.

During program compression, CICS frees the storage of all programs not currently in use and marks for deletion programs that are loaded so low in the DSA that they are mixed in with, or are near, nonprogram storage. Some program compression is a natural result of the design of CICS, since non-resident programs are not automatically removed from the DSA at the end of a task. (CICS does not automatically remove non-resident programs in case a subsequent tasks would require the program.) Since the non-resident programs are not automatically removed, DSA will periodically become full and program compression must be implemented to free the storage.

If the GETMAIN causing program compression to be invoked cannot be satisfied after program compression, CICS enters a "short-on-storage" condition. A shortage of free space in the dynamic storage area is a serious performance problem for a CICS region. This situation is referred to as CICS entering a "stress" condition and a message is sent to the operator ("DHF0506 CICS/VS IS UNDER STRESS - SHORT-ON-STORAGE")

After the short-on-storage is signaled, CICS stops various activities that might require dynamic storage. For example, CICS generally stops creating new tasks and stops inviting input messages to start new tasks. These actions translate into poor performance for the end user.

Additionally, since CICS stops inviting input messages until the stress condition is alleviated, a "backlog" of input messages can become artificially queued at the terminals. This is because an increasing number of users may be waiting for CICS to accept their inputs. Consequently, when the short-on-storage condition is relieved, many input messages may be waiting to be submitted. These transactions would occur in "burst" mode, and may cause additional CICS stress conditions!

The CICS statistics report the number of times CICS signaled a short-on-storage condition because available dynamic storage had fallen below a threshold limit or was too fragmented for a large area to be allocated (reported as "Number of cushion releases"). Additionally, the CICS report the number of times a new storage queue was started (reported as "Times requests queued from zero"). The sum of the number of cushion releases and the times a new storage queue was started represents the total number of times that CICS had a short-on-storage condition, meaning that there was a shortage of free space in the dynamic storage area.

CPExpert sums the "Number of cushion releases" and the "Times requests queued from zero" values. CPExpert produces Rule CIC110 if the result is greater than zero.

Suggestion: CPExpert strongly suggests that you investigate the causes of the short-on-storage situation and take actions to alleviate the situation. You should increase the size of the dynamic storage area or tune CICS to minimize the demands on the dynamic storage area.

There are too many possible ways to minimize the demands on the dynamic storage area to describe in this Rule Description (for example, page 187 of the CICS Performance Guide for CICS 1.7 lists 51 actions to improve usage of virtual storage). Furthermore, including the actions in this document would unnecessarily duplicate the excellent guidance in IBM's CICS Performance Guide.

CPExpert suggests that you review the suggested actions in the IBM CICS Performance Guide appropriate to your version of CICS.

- CICS/OS/VS Version 1.7 Performance Guide: page 187.
- CICS/MVS Version 2.1.2 Performance Guide: page 227.
- CICS/ESA Version 3.1.1 Performance Guide: page 203.
- CICS/ESA Version 3.2 Performance Guide: page 107.
- CICS/ESA Version 3.3 Performance Guide: page 117.
- CICS/ESA Version 4.1 Performance Guide: Section 4.1.2
- *CICS/TS Release 1.1 Performance* Guide: Section 4.1.2

- *CICS/TS Release 1.2 Performance* Guide: Section 4.1.2
- *CICS/TS Release 1.3 Performance* Guide: Section 4.1.2
- CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.1.2 |
- CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.1.2 |

When Rule CIC110 is produced, CPExpert will attempt to determine which of the possible actions listed in the CICS Performance Guide could result in a savings of storage. CPExpert reports the results of its analysis by producing different rules associated with potential actions.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 155 and 339.

CICS/MVS Version 2.1.2 Performance Guide: pages 87 and 278.

CICS/ESA Version 3.1.1 Performance Guide: pages 173 and 319.

CICS/ESA Version 3.2.1 Performance Guide: pages 78, 216, and 330.

CICS/ESA Version 3.3.1 Performance Guide: pages 88, 214, and 349.

CICS/ESA Version 4.1 Performance Guide: Section 4.7.5 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 3.2.3.2, Section 4.7.5, and Appendix 1.1.27.

CICS/TS Release 1.2 Performance Guide: Section 3.2.3.2, Section 4.1.2 and Appendix 1.1.24.

CICS/TS Release 1.3 Performance Guide: Section 3.2.3.2, Section 4.1.2 and Appendix 1.1.25.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 11 (Short-on-storage condition), Part 4 (Virtual storage above and below 16MB line checklist), and Appendix A (Table 128).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 3.2.2.2 (Short-on-storage condition), Section 4.1.2 (Virtual storage above and below 16MB line performance checklist) |

Rule CIC112: The number of GETMAINs is high

Finding: CPExpert has detected that the number of GETMAINs issued by tasks is higher than should normally occur.

Impact: This finding normally has a LOW IMPACT on the performance of the CICS region, although the finding could have a MEDIUM IMPACT or HIGH IMPACT on the performance of individual CICS tasks. Additionally, excessive GETMAINs generate unnecessary overhead[1].

Discussion: The CICS statistics report the number of times CICS acquired storage for a task (CICS issued a GETMAIN). The number of GETMAINs is related to the design and use of the tasks and applications. Consequently, there are no CICS tuning controls directly related to the number of GETMAINs.

GETMAINs impose system overhead for the system code necessary to execute the GETMAIN. It is desirable to minimize the GETMAINs per task to minimize this overhead, although tasks obviously must issue GETMAINs to acquire any necessary storage. As a "rule of thumb," there should be fewer than an average of 25 GETMAINs per task[2]. More than this value generally indicates that the task may be improperly designed or that it is experiencing performance problems of some type.

It is not uncommon to see hundreds or even thousands of GETMAINs executed by poorly designed (or poorly implemented) tasks!

CPExpert divides the number of storage acquisitions by the number of tasks. The result from this division is an average; some tasks would have issued relatively few GETMAINs, while other tasks would have issued many more than the average. CPExpert produces Rule CIC112 if the result is greater than the GETMAIN guidance variable in USOURCE(CICGUIDE). The default value for the GETMAIN guidance variable is 25, so CPExpert will produce Rule CIC112 when the average task in a CICS region issues more than 25 GETMAINs .

CPExpert reports the average number of GETMAINs resulting from the above computation. When evaluating this value, keep in mind that the

---

[1] The “Sample Performance Data” of IBM *CICS Performance Guides* show that each GETMAIN and associated FREEMAIN require almost 2 milliseconds of CPU time.

[2] The IBM *CICS Performance Guides* specify that storage control problems would be indicated by “a very high number of storage control requests per task (perhaps 50 or 100)” but a reasonable limit on the number of GETMAINs would typically be less than 25 per task.

value represents an average over all tasks. Some tasks will issue very few GETMAINs, which means that some tasks will issue many more than the average.

Suggestion: CPExpert suggests that you investigate the abnormal number of GETMAINs. If you conclude that the number of GETMAINs per task is excessive, consider the following alternatives:

- Discuss the tasks with system developers to determine whether the number of GETMAINs can be reduced, or whether the number of GETMAIN requests consistent with the number and types of tasks.

  Once a problem of this type has been brought to the attention of system developers, they often can quickly resolve the situation. As examples of problems that system developers might identify:

  - A GETMAIN could be enclosed within a loop, so the GETMAIN is executed with each iteration of the loop. Simply moving the GETMAIN outside the loop could dramatically reduce the number of GETMAINs.

  - A task might be issuing GETMAIN requests repetitively, each for a reasonable amount of storage, but collectively for a very large amount of storage. Performance can be improved by obtaining all required storage in one GETMAIN rather than several smaller GETMAINs.

  - The IOAREALEN (value1) or TIOAL might be smaller than most initial terminal inputs. In this case, excessive GETMAIN requests can occur, resulting in additional processor requirements (unless IOAREALEN(value1) or TIOAL is zero). In general, a value of zero is best because it causes the optimum use of storage and eliminates the second GETMAIN request.

  - In a MRO environment, IOAREALEN might be smaller than most messages transmitted on the MRO link. In this case, excessive FREEMAIN and GETMAIN requests can occur, resulting in additional processor requirements.

  - Language Environment runtime options can have a major impact on both storage usage and CPU costs of Java application programs running under CICS. The key Language Environment runtime options for Java application programs are STACK, HEAP and ANYHEAP. If the initial size for any of these options is too large, excessive storage will be allocated, which may result in a short-on-storage condition in the CICS region. If an initial value is too small, Language Environment will issue a GETMAIN to allocate additional storage, which increases the CPU cost. Additional CPU cost can also be incurred due to extra GETMAINs and FREEMAINs if the FREE parameter is specified for any option where the initial size is too small.

- The system pathlength increases when a CICS application invoked by Language Environment issues an EXEC CICS LINK request. Repeated EXEC CICS LINK calls to the same program invoked by Language Environment result in *multiple* GETMAIN/FREEMAIN requests for run-time unit work areas (RUWAs).

  RUWAPOOL(YES) results in the creation of a run_unit work area pool during task initialization. This pool is used to allocate RUWAs required by LE_conforming programs. This reduces the number of GETMAINS and FREEMAINS in tasks which perform many EXEC CICS LINKS to LE_conforming programs.

- When the Java program object is invoked in a CICS system, a storage report can be written[3] to the CICS CESE transient data destination (usually directed to the data set defined by the CEEMSG DD statement). The report shows the number of system level get storage calls, such as EXEC CICS GETMAIN, that were required while the application was running. To improve performance, use the storage report numbers as an aid in setting the initial and increment size for STACK, HEAP and ANYHEAP to values which will reduce the number of times that the language environment storage manager makes requests to acquire storage.

- If necessary, analyze the task issuing an unacceptable number of GETMAINs in some detail. This can be done by performing "single-transaction measurement" as outlined in the CICS Performance Guide.

- You can change the **GETMAIN** guidance variable in USOURCE(CICGUIDE) if you feel that Rule CIC112 is produced prematurely. Section 3 describes how to change the GETMAIN guidance variable.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 50 and 171.

CICS/MVS Version 2.1.2 Performance Guide: pages 108 and 422.

CICS/ESA Version 3.1.1 Performance Guide: pages 117 and 192.

[3]To get a report on the storage used by your Java program object, specify the following runtime option on the hpj command _lerunopts="RPTSTG(ON)". RPTSTG **should only be used in a test environment** because of the overheads incurred in writing the storage report each time the Java program object is executed.

CICS/ESA Version 3.2.1 Performance Guide: pages 96 and 331.

CICS/ESA Version 3.3.1 Performance Guide: pages 106 and 351.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.3.4.

CICS/TS Release 1.1 Performance Guide: Section 3.3.4.

CICS/TS Release 1.2 Performance Guide: Section 3.3.4.

CICS/TS Release 1.3 Performance Guide: Section 3.3.4.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 3.1.2, Section 4.3.1 (Setting the size of the terminal input/output area), *Section* 4.6.3.2 (Language Environment runtime options)

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 3.1.2, Section 4.3.1 (Setting the size of the terminal input/output area), *Section* 4.6.3.2 (Language Environment runtime options)

Also, please review the “Loops” section of the *CICS Problem Determination Guide* for your version of CICS.

Rule CIC113: Peak EDSA usage is approaching EDSALIM value

Finding: CPExpert has detected that the peak usage of the Extended Dynamic Storage Area (EDSA) is approaching the maximum amount specified by the EDSALIM keyword in the System Initialization Table.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The value of the EDSALIM keyword in the System Initialization Table (SIT) specifies the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside above the 16MB boundary. CICS allocates the following extended dynamic storage areas from the storage value that you specify on the EDSALIM parameter:

- The **Extended User DSA (EUDSA)**, which is the user-key storage area for all user-key task-lifetime storage above the 16MB boundary.
- The **Extended Read-only DSA (ERDSA)**, which is the key-0 storage area for all reentrant programs and tables above the 16MB boundary.
- The **Extended Shared DSA (ESDSA)**, which is the user-key storage area for any non-reentrant user-key RMODE(ANY) programs, and also for any storage obtained by programs issuing CICS GETMAIN commands for storage above the 16MB boundary with the SHARED option.
- The **Extended CICS DSA (ECDSA)**, which is the CICS-key storage area for all non-reentrant CICS-key RMODE(ANY) programs, all CICS-key task-lifetime storage above the 16MB boundary, and CICS control blocks that reside above the 16MB boundary.

The default EDSA limit is 20MB (20,971,520 bytes). The maximum value allowed depends on a number of factors, such as:

- The size of the region specified on the MVS REGION parameter in the CICS job or procedure.
- How much storage is required for the CICS internal trace table.

- How much private storage that must be left free to satisfy MVS GETMAIN requests for storage above the 16MB boundary outside the Dynamic Storage Areas.

Too small a dynamic storage area for the workload being processed by the CICS region results in increased program compression, SOS (short on storage) conditions, or storage deadlock ABENDS when program compression is not sufficient. None of these results are desired, and they particularly are not desired in an important CICS region. Consequently, it can be important to be aware of the percent of EDSA that is used.

The CICS interval statistics contain information about the amount of EDSA allocated. CPExpert compares the peak EDSA allocated against a percent of the value specified for the EDSALIM keyword. CPExpert produces Rule CIC113 when the peak EDSA allocated storage is greater than this percent. The percent used by CPExpert is specified by the **PCTEDSA** guidance variable in USOURCE(CICGUIDE).

Suggestion: Rule CIC113 is intended to alert you to a potential problem, in that the peak allocated EDSA is approaching the limit specified by the EDSALIM keyword in the SIT. CPExpert suggests that you consider the following alternatives:

- Determine whether the default value of the PCTEDSA guidance variable in USOURCE(CICGUIDE) is appropriate for the CICS region being analyzed. Please alter the PCTEDSA guidance variable if appropriate.

- Determine whether the value of the EDSALIM keyword is appropriate. If possible, consider increasing the value of the EDSALIM keyword.

- Take action to reduce the amount of storage required by the CICS region. CPExpert suggests that you review the suggested actions in the "**Virtual storage above and below 16MB line checklist**" in the IBM CICS Performance Guide appropriate to your version of CICS.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 155 and 339.

CICS/MVS Version 2.1.2 Performance Guide: pages 87 and 278.

CICS/ESA Version 3.1.1 Performance Guide: pages 173 and 319.

CICS/ESA Version 3.2.1 Performance Guide: pages 78, 216, and 330.

CICS/ESA Version 3.3.1 Performance Guide: pages 88, 214, and 349.

CICS/ESA Version 4.1 Performance Guide: Section 4.7.6.8 and Appendix A.1.30.

CICS/TS Release 1.1 Performance Guide: Section 4.7.6.8 and Appendix 1.1.27, and Appendix 6.6.

CICS/TS Release 1.2 Performance Guide: Section 4.7.7.8, Appendix 1.1.24, and Appendix 6.6.

CICS/TS Release 1.3 Performance Guide: Section 4.11.6.8, Appendix 1.1.25, and Appendix 6.6.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 23 (Extended dynamic storage areas) and Appendix F (Dynamic Storage Areas).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.10.7.1 (Extended dynamic storage areas) and APPENDIX1.1.25.4 Storage manager: Dynamic storage areas statistics.

Thanks: Thanks to Paul Gordon (NationsBank) for suggesting this rule.

Rule CIC120: A storage violation occurred in a production CICS region

Finding: CPExpert has detected that a storage violation occurred in a production CICS region.

Impact: This finding should normally have a HIGH IMPACT on the performance of the CICS region, since storage violations should never occur in a production CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: Storage violations are detected by the CICS storage control program. A storage violation can occur in two basic situations:

- When the CICS storage manager detects an error during its normal processing of a GETMAIN or FREEMAIN request.

- When a program check occurs and the system recovery process determines that the storage manager was in control at the time. The storage manager deliberately references storage when it is following chains of allocated or free areas. This is done to force a program check if the pointer is bad.

Storage violations can be categorized as minor or severe. Minor storage violations are ones that can be recovered and usually involve only one storage area. Severe storage violations are ones that cannot be recovered. They usually are the result of many areas being corrupted in such a way that recovery is unable to rebuild the chains. If the recovery procedures are not successful, CICS is terminated and must be reinitialized.

Any storage violation is costly; it involves processor overhead and page faults. If storage violations occur in a production CICS region, the result may be that the region must be terminated and restarted. Such an occurrence would, of course, have serious performance implications.

CPExpert produces Rule CIC120 if any storage violation is reported for a production region.

Suggestion: CPExpert suggests that you immediately investigate and correct the causes of the storage violations.

CPExpert will identify the task **experiencing** the storage violation if data is available (that is, if CICS shutdown statistics are available for analysis or if the performance data base retains the information).

However, the task experiencing the storage violation may not be the task **causing** the storage violation. This is because one task may damage the storage chain and not experience a storage violation. The storage violation may not be experienced until another task references the storage chain.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 51 and 160.

CICS/MVS Version 2.1.2 Performance Guide: pages 91 and 421.

CICS/ESA Version 3.1.1 Performance Guide: pages 118 and 176.

CICS/ESA Version 3.2.1 Performance Guide: pages 80 and 331.

CICS/ESA Version 3.3.1 Performance Guide: pages 90 and 303.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.6 and Appendix A.1.29.

CICS/TS Release 1.1 Performance Guide: Section 3.2.6 and Appendix 1.1.29

CICS/TS Release 1.2 Performance Guide: Section 3.2.6 and Appendix 1.1.29.

CICS/TS Release 1.3 Performance Guide: Section 3.2.6 and Appendix 1.1.32.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 11 (Recovery from storage violation) and Appendix A (Table 111).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 3.2.4 (Detecting storage violation) and APPENDIX1.1.25.

Rule CIC121: Transactions experiencing storage violations

Finding: CPExpert detected that a storage violation occurred in a production CICS region. This rule lists the transactions involved.

Impact: This finding should normally have a HIGH IMPACT on the performance of the CICS region, since storage violations should never occur in a production CICS region.

Logic flow: The following rule causes this rule to be invoked:

Rule CIC120: A storage violation occurred

Discussion: Storage violations are detected by the CICS storage control program. A storage violation can occur in two basic situations:

- When the CICS storage manager detects an error during its normal processing of a GETMAIN or FREEMAIN request.
- When a program check occurs and the system recovery process determines that the storage manager was in control at the time. The storage manager deliberately references storage when it is following chains of allocated or free areas. This is done to force a program check if the pointer is bad.

CPExpert produces Rule CIC120 if any storage violation is reported for a production region.

Suggestion: CPExpert will identify the task **experiencing** the storage violation if data is available (that is, if CICS shutdown statistics are available for analysis or if the performance data base retains the information).

However, the task experiencing the storage violation may not be the task **causing** the storage violation. This is because one task may damage the storage chain and not experience a storage violation. The storage violation may not be experienced until another task references the storage chain.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 51 and 160.

CICS/MVS Version 2.1.2 Performance Guide: pages 91 and 421.

CICS/ESA Version 3.1.1 Performance Guide: pages 118 and 176.

CICS/ESA Version 3.2.1 Performance Guide: pages 80 and 331.

CICS/ESA Version 3.3.1 Performance Guide: pages 90 and 303.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.6 and Appendix A.1.29.

CICS/TS Release 1.1 Performance Guide: Section 3.2.6 and Appendix 1.1.29

CICS/TS Release 1.2 Performance Guide: Section 3.2.6 and Appendix 1.1.29.

CICS/TS Release 1.3 Performance Guide: Section 3.2.6 and Appendix 1.1.32.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 11 (Recovery from storage violation) and Appendix A (Table 111).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 3.2.4 (Detecting storage violation) and APPENDIX1.1.25.

Rule CIC130: Storage dumps occurred for a production CICS region

Finding: CPExpert has detected that storage dumps were taken for a production CICS region.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region, since the dumps should occur infrequently. However, the performance impact could be significant if the number of dumps is large. In a heavily-loaded system, the performance impact may be large even if the number of dumps is relatively small. This is because CICS tasks get backlogged and CICS could become stressed recovering from the backlog.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: Storage dumps are produced for a variety of reasons (e.g., program checks and storage violations). The system is severely loaded while writing a dump to the CICS dump data set. In the case of formatted dumps, nothing else is processed by CICS until the dump is written to the CICS dump data set. Depending upon the size of the dump, the dump could take over one second. During this time, all other activities are delayed. All storage dumps in a production CICS region should be thoroughly investigated and their causes eliminated.

CPExpert produces Rule CIC130 if the number of storage dumps is greater than the STORDUMP guidance variable in USOURCE.CICGUIDE.

Suggestion: CPExpert suggests that you investigate and correct the causes of the storage dumps.

- If the CICS transactions causing the storage dumps are in production mode, additional quality control may be indicated.

- If application developers are testing new CICS applications in the region, perhaps a CICS test region should be established for the test work.

In any event, storage dumps are disruptive to production CICS regions. The more heavily-loaded the CICS region, the more disruptive the effect of the storage dumps. Significant effort should be expended to find and correct the causes of the storage dumps.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 56.

CICS/MVS Version 2.1.2 Performance Guide: pages 238 and 376.

CICS/ESA Version 3.1.1 Performance Guide: page 63.

CICS/ESA Version 3.2.1 Performance Guide: pages 48 and 283.

CICS/ESA Version 3.3.1 Performance Guide: pages 50 and 302.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.6 and Appendix A.1.6.

CICS/TS Release 1.1 Performance Guide: Section 3.2.6 and Appendix 1.1.6

CICS/TS Release 1.2 Performance Guide: Section 3.2.6 and Appendix 1.1.6

CICS/TS Release 1.3 Performance Guide: Section 3.2.6 and Appendix 1.1.6.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 11 (Recovery from storage violation) and Appendix A (Table 111).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 3.2.4 (Detecting storage violation) and APPENDIX1.1.25. |

Rule CIC140: The number of transaction errors is high

Finding: CPExpert has determined that the number of transaction errors is high for the indicated terminal.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region. However, the finding potentially can have a HIGH IMPACT if the finding results from an attempt to bypass security controls.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: Transaction errors are errors in which the transaction associated with a particular terminal could not be started. This means that:

- A transaction identifier had not been defined in the CICS System Definition (CSD) data set or in the Program Control Table (DFHPCT).
- The operator did not have the proper security access to enter the transaction.
- The transaction had been disabled.

CPExpert produces Rule CIC140 when the number of transaction errors for any particular terminal exceeds the value specified for the TRANSERR guidance variable in USOURCE.CICGUIDE.

Suggestion: CPExpert suggests that you determine why the terminal operator at the indicated terminal caused the transaction errors.

- The transaction identifier might not have been properly defined in the CSD data set or in the PCT. This situation is unlikely to occur often, since the problem would be immediately apparent and would be quickly corrected.
- The terminal operator might not have the proper security to enter the transaction. This situation could be serious, since it could indicate that attempts are being made to bypass security controls.
- The terminal operator may have entered the transaction identifier in error. Additional operator training may be indicated if the application system is new or if the operator is inexperienced.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 58.

CICS/MVS Version 2.1.2 Performance Guide: page 385.

CICS/ESA Version 3.1.1 Performance Guide: page 134.

CICS/ESA Version 3.2.1 Performance Guide: page 344.

CICS/ESA Version 3.3.1 Performance Guide: page 364.

CICS/ESA Version 4.1.1 Performance Guide: Appendix A.1.28.

CICS/TS Release 1.1 Performance Guide: Appendix 1.1.26.

CICS/TS Release 1.2 Performance Guide: Appendix 1.1.27.

CICS/TS Release 1.3 Performance Guide: Appendix 1.1.30.

CICS/TS for z/OS Release 2.1 *Performance Guide*:Appendix A (Table 126).

CICS/TS for z/OS Release 2.1 *Performance Guide*: APPENDIX 1.1.30.2 |

Rule CIC141: Transactions experienced a stall-purge

Finding: CPExpert has determined that transactions experienced a stall-purge in a production CICS region.

Impact: This finding should normally have a VERY HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: When CICS is in a short-on-storage condition or the maximum-task limit (specified by the MXT keyword in the SIT) has been reached, CICS will not attach new tasks.

In some situations, a "deadlock" situation can occur in which no task is able to proceed. This situation can arise because tasks are waiting on the completion of other tasks, but the other tasks cannot proceed because of storage limits, or because of AMXT or MXT limits. CICS considers this to be a **system stress** situation, and takes action to resolve the situation.

CICS attempts to solve the deadlock situation by temporarily raising the AMXT (active maximum tasks) value. A task may be causing a deadlock because the number of active tasks in the CICS region has reached the region's AMXT value and the task cannot execute. Potentially, temporarily increasing the AMXT value would allow the task to execute and free the deadlock.

If no tasks can proceed after raising the AMXT value, CICS waits for an interval to see if the deadlock will be resolved. (The deadlock could be resolved by the completion of some event. For example, the task causing the deadlock could be enqueued on some resource or waiting for terminal input.) The interval CICS will wait to see if the deadlock is resolve is a parameter provided by the ICVS keyword in the SIT (the default value for the ICVS keyword is 20 seconds).

After the expiration of the ICVS interval, CICS will attempt to remove the deadlock by purging tasks. CICS will begin purging those tasks which have been defined as SPURGE=YES in DFHPCT or CEDA DEFINE transaction. (The SPURGE keyword indicates whether the transaction can be purged in a CICS stall condition. The default is SPURGE=NO.) CICS will purge one transaction, selecting the lowest priority task. CICS will then wait another ICVS interval to see if the deadlock is removed.

If all transactions with SPURGE=YES are purged and no task is able to proceed, it is necessary to cancel CICS and restart the CICS region.

It is not too uncommon for stall conditions to occur in a test CICS region. This is because a stall condition may be the result of transaction interactions, and the task or system logic may not be completely debugged. **However, a system stall condition should never occur in a production CICS region**. Any purge-stalls of transactions in a production CICS region indicates the need for immediate attention to determine the cause and correct the problem.

An additional situation can arise in which individual tasks may be stalled (or deadlocked), waiting on some action by other tasks. CICS takes no action in this situation if any task in the CICS region is able to proceed. No information is provided about this situation, since CICS does not view the situation as creating a system stress condition.

This situation can be detected based upon user complaints or observant operators (perhaps using a commercially-available CICS monitoring tool). These situations can be removed by issuing a CEMT SET TASK PURGE command. However, these situations should not normally occur in a production CICS region. Remedial programming or system design should be focused on the transactions if these situations occur in a production CICS region.

CPExpert produces Rule CIC141 if any transactions were stall-purged in a production CICS region. CPExpert suppresses this rule for CICS Version 3 since data are not available.

Suggestion: CPExpert suggests that you take immediate action to determine the cause of the stall-purges.

- CPExpert will produce other rules if the CICS region has encountered a short-on-storage condition (for example, Rule CIC110 will be produced). The advice provided by these rules should be followed.

- If CPExpert does not detect a short-on-storage condition, the system stress condition is caused by such problems as system design logic, enqueue conflicts, task logic errors, operator responsive, etc. You should determine the cause of the system stress and take remedial action.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 53 and 156.

CICS/MVS Version 2.1.2 Performance Guide: pages 89 and 381.

CICS/ESA Version 3.1.1 Performance Guide: pages 116 and 174.

CICS/ESA Version 3.2.1 Performance Guide: pages 78 and 330.

CICS/ESA Version 3.3.1 Performance Guide: pages 88 and 349.

CICS/ESA Version 4.1.1 Performance Guide: Section3.2.3 and Appendix A.1.24.

CICS/TS Release 1.1 Performance Guide: Section 3.2.3.3 and Appendix 1.1.23.

CICS/TS Release 1.2 Performance Guide: Section 3.2.3.3 and Appendix 1.1.24.

CICS/TS Release 1.3 Performance Guide: Section 3.2.3.3 and Appendix 1.1.25.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 11 (Purging of Tasks) and Appendix A (Table 111).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Sectoin 3.2.2.3 Purging of tasks and APPENDIX 1.1.31.1.

Rule CIC150: VTAM reached the maximum RAPOOL value too often

Finding: CPExpert has determined that VTAM reached the maximum Receive-Any Pool (RAPOOL) too often.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: Task input from a terminal is received by the VTAM access method and is passed to CICS, if CICS has a receive-any request outstanding. CICS receives data from VTAM in the receive-any input area (RAIA). The RAMAX operand in the SIT specifies the size in bytes of the I/O area that is allocated for each VTAM receive-any operation.

The RAMAX value should generally be set as large as the normal CICS system input message. For example, if the normal CICS system input is a screen, the RAMAX should be set to the size of the screen (plus 23 bytes to hold storage accounting information added by CICS).

The RAPOOL operand in the System Initialization Table (SIT) specifies the number of concurrent receive-any requests that CICS is to process from VTAM. The value of the RAPOOL operand determines how may receive-any buffers are created at any time. The number of buffers determines how many simultaneous VTAM inputs will be handled directly by CICS buffers, rather than by VTAM buffers.

If the number of simultaneous VTAM inputs is larger than the RAPOOL value, VTAM will store the inputs into its own buffers (acquired from subpool 229) and transfer them to CICS when CICS buffers are available.

If RAPOOL is set too low, not all terminal messages may be processed during one dispatch of the terminal control program. This will cause transactions to be delayed and performance to suffer. If RAPOOL is set too high (and RAMAX is a large value), then storage is wasted.

For each receive-any request, a VTAM request parameter list (RPL), a receive-any control element (RACE), and a receive-any input area are reserved. The total area reserved for a VTAM receive-any operation is:

(RAMAX value + RACE size + RPL size) * RAPOOL value

If virtual storage is not constrained, if is wise to have RAPOOL high enough to accommodate all receive-any requests. However, if storage is constrained, there are tradeoffs involved.

There are three basic considerations in selecting the values for RAMAX and RAPOOL: (1) the amount of storage required to handle the inputs, (2) the amount of processor time required to reissue the receive-any requests if inputs cannot be completely transferred to CICS, and (3) the delays to CICS tasks because their input cannot be serviced when available to VTAM.

CPExpert evaluates the "number of times reached maximum" in the VTAM statistics portion of the CICS statistics. Rule CIC160 is produced if this value is greater than the RAPOOL guidance variable and storage was not constrained. (The default for the RAPOOL guidance variable is zero.)

Suggestion: CPExpert suggests that you consider the following alternatives:

- Increase the RAPOOL value in the SIT. This will allow more concurrent inputs to be processed by VTAM in the CICS buffers. The RAPOOL value generally should be increased by 2 or 3 over its present setting.

- Assess the CICS workload executing during the interval being analyzed by CPExpert. If the workload is unusual (e.g., end of month processing or some other relatively infrequent processing) then you may wish to ignore this rule. It may be better to allow VTAM to store the peak of messages in its own areas acquired from subpool 229, than to allocate more space to CICS which would normally be unused.

- Assess the CICS application systems - there may be unique situations in which CICS reissues a receive-in as soon as it finds one satisfied. In this situation, CICS would use the same RAIA repeatedly, and additional areas would be of no advantage. If the RAMAX size is relatively small, it is not worth spending much time trying to determine whether this situation occurs since having a RAPOOL value larger than necessary would "waste" only a small amount of storage.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 59-60 and pages 209-213.

CICS/MVS Version 2.1.2 Performance Guide: pages 131-133 and pages 387-388.

CICS/ESA Version 3.1.1 Performance Guide: pages 144-146 and pages 209-213.

CICS/ESA Version 3.2.1 Performance Guide: pages 127-131 and pages 354-355.

CICS/ESA Version 3.3.1 Performance Guide: pages 137-141 and pages 372-373.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.3.2, Section 4.3.3, and Appendix A.1.35.

CICS/TS Release 1.1 Performance Guide: Section 4.3.2, Section 4.3.3, and Appendix 1.1.31.

CICS/TS Release 1.2 Performance Guide: Section 2.2.14, and Appendix 1.1.32.

CICS/TS Release 1.3 Performance Guide: Section 2.2.14, and Appendix 1.1.35.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (VTAM statistics) and Appendix A (Table 148).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.3.3 Setting the size of the receive-any pool (RAPOOL).

Rule CIC151: The RAPOOL value may be too large

Finding: CPExpert has determined that the value specified for the Receive-Any Pool (RAPOOL) may be too large.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: The following rule caused this rule to be invoked:
Rule CIC110: CICS encountered a Short-on-Storage condition

Discussion: The discussion associated with Rule CIC150 describes how the RAPOOL value is used to control the number of concurrent CICS receive-any requests. Please refer to that rule for basic discussion.

The VTAM request parameter list (RPLs) is used to satisfy input requests from VTAM terminals to start transactions. CICS provides a pool of RPLs using the value specified in the RAPOOL operand.

The CICS requested and shutdown VTAM statistics show the maximum number of RPLs posted by VTAM on any one dispatch of terminal control.

The maximum number of RPLs is less than the value specified for the RAPOOL operand in the data analyzed by CPExpert. This means that the RAPOOL operand is too large. If the RAPOOL value is too large, the excess virtual storage is wasted. (The RAPOOL value does not seriously affect real storage because the storage is not page-fixed and can be paged out.)

However, there is little damage caused by the excessive value in the RAPOOL operand if virtual storage is not constrained.

Suggestion: CPExpert suggests that you consider decreasing the RAPOOL value in the SIT. The RAPOOL value generally should be decreased to the value shown in the maximum number of RPLs.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 59-60 and pages 209-213.

CICS/MVS Version 2.1.2 Performance Guide: pages 131-133 and pages 387-388.

CICS/ESA Version 3.1.1 Performance Guide: pages 144-146 and pages 209-213.

CICS/ESA Version 3.2.1 Performance Guide: pages 127-131 and pages 354-355.

CICS/ESA Version 3.3.1 Performance Guide: pages 137-141 and pages 372-373.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.3.2, Section 4.3.3, and Appendix A.1.35.

CICS/TS Release 1.1 Performance Guide: Section 4.3.2, Section 4.3.3, and Appendix 1.1.31.

CICS/TS Release 1.2 Performance Guide: Section 2.2.14, and Appendix 1.1.32.

CICS/TS Release 1.3 Performance Guide: Section 2.2.14, and Appendix 1.1.35.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (VTAM statistics) and Appendix A (Table 148).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.3.3 Setting the size of the receive-any pool (RAPOOL).

Rule CIC155: The number of concurrent autoinstall terminals may be too small

Finding: CPExpert has determined that the number of concurrent Autoinstall terminals may be too small.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The Automatic Installation of Terminals (Autoinstall) is controlled by the AUTINST operand in the SIT (for CICS versions before Version 3) and the AIQMAX operand in the SIT (for CICS Version 3).

During Autoinstall processing, CICS will obtain storage to handle each Autoinstall request. This storage is obtained from the control subpool in the Dynamic Storage Area (DSA). The amount of virtual storage obtained will be between 120-250 bytes for a typical Autoinstall request.

The Autoinstall task (CATA) is the principal consumer of CICS resources in the Autoinstall process.

Unless there is a virtual storage constraint, the value specified for the maximum number of concurrent Autoinstall terminals (the AUTINST or AIQMAX operands) should be greater than the number of Autoinstall attempts.

However, if there is a virtual storage constraint or if the CICS region becomes overloaded by the number of concurrent autoinstall tasks, the AUTINST or AIQMAX operands can be used to limit the amount of resources required by the Autoinstall process.

CPExpert did not detect a virtual storage constraint in the data being analyzed. However, the "peak concurrent attempts" in the statistics was as large as the number of concurrent Autoinstall attempts specified in the SIT, and there were some rejected Autoinstall attempts.

Suggestion: CPExpert suggests that you consider increasing the number of concurrent autoinstall terminals by increasing the value of the AUTINST operand for CICS/XA; or the AIQMAX operand for CICS/ESA, CICS/TS for OS/390, or CICS/TS for z/OS.

Of course, you may have previously experienced problems caused by a large number of devices queued for autoinstall. If this should be the case, perhaps no adjustment of the number of autoinstall terminals is appropriate.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/ESA Version 3.1.1 Performance Guide*, page 43 and pages 224-226.

CICS/ESA Version 3.2.1 Performance Guide, pages 141-143 and page 265.

CICS/ESA Version 3.3.1 Performance Guide, pages 151-154 and page 285.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.3.11 and Appendix A.1.1.

CICS/TS Release 1.1 Performance Guide: Section 4.3.11 and Appendix 1.1.1.

CICS/TS Release 1.2 Performance Guide: Section 4.3.3 and Appendix 1.1.1.

CICS/TS Release 1.3 Performance Guide: Section 4.3.11 and Appendix 1.1.2.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 16 (Automatic installation of terminals) and Appendix A (Table 16).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.3.11 (Tuning automatic installation of terminals) and APPENDIX 1.1.30.1.

Rule CIC156: The DSHIPINT value might be too large

Finding: CPExpert believes that the value for the DSHIPINT parameter may be too large.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region, but the impact could be LARGE during times when the CICS-supplied CRMF transaction executes This finding applies to CICS Release 4.1 and later releases of CICS.

Logic flow: This is a basic finding based on analysis of the CICS interval statistics.

Discussion: In a transaction routing environment, terminal definitions can be "shipped" from a terminal-owning region (TOR) to an application-owning region (AOR). The terminal definition becomes redundant in the AOR under the following conditions:

- The terminal user logs off.
- The terminal user stops using transactions which route to the AOR.
- The TOR on which the user is signed on is shut down.
- The TOR is restarted without recovering autoinstalled terminal definitions, and the autoinstall user program DFHZATDX assigns a new set of terminal ids to the same set of terminals.

Shipped terminal definitions which have become redundant may need to be deleted. Long-lasting shipped terminal definitions do not generally cause storage problems because of the relatively small amounts of storage which they occupy. However, there are other considerations, such as security, which may require that redundant shipped terminal definitions are not allowed to persist in an AOR.

The CICS-supplied transaction CRMF periodically scans the shipped terminal definitions in the AOR and flags those which it has determined to be redundant. If any redundant definitions have been identified, the CICS-supplied transaction CRMD is invoked to delete the redundant definitions. This processing is referred to as the CICS timeout delete mechanism.

The system initialization parameters DSHIPINT and DSHIPIDL control the amount of time for which a redundant shipped terminal definition is allowed to survive and the frequency at which shipped terminal definitions are tested for redundancy.

- The DSHIPINT parameter in the SIT specifies the interval between invocations of the CRMF transaction. The CRMF transaction searches for shipped transactions have not been used longer than that specified by the DSHIPIDL parameter. The default value for the DSHIPINT parameter is 12 hours, meaning that the CRMF transaction will be invoked every 12 hours to search for redundant shipped terminal definitions.

- The DSHIPIDL parameter in the SIT specifies the minimum time, in hours, minutes, and seconds, that an inactive shipped terminal definition must remain installed in this region. When the CRMF transaction is invoked, only those shipped definitions that have been inactive for longer than the specified time are deleted. The default value for the DSHIPIDL parameter is 2 hours, meaning that shipped terminal definitions will be considered redundant if they have been unused for more than 2 hours.

Both of these parameters can be altered dynamically by issuing a CEMT SET DELETSHIPPED or EXEC CICS SET DELETSHIPPED command to reset the intervals.

If the CRMF/CRMD processing deletes a shipped terminal definition, the terminal definition must be re-shipped when the terminal user next routes a transaction from the TOR to the AOR. The deletion of the terminal definition and re-shipping incurs CPU overhead that can be eliminated or minimized with appropriate settings of the DSHIPINT and DSHIPIDL parameters.

The IBM *CICS Performance Guides* indicate that the default setting of 2 hours for the DSHIPIDL parameter is a reasonable value, unless security issues dictate a lower value. However, the default setting of DSHIPINT might not be appropriate.

If the default value of 12 hours (or a similarly high value) is specified for the DSHIPINT parameter, the CRMF transaction might identify a considerable number of redundant terminal definitions during a single expiration of the DSHIPINT value. Consequently, a relatively large "burst" of CPU would be required for the CRMD transaction processing.

CPExpert compares the A04SKDEL variable in CICSAUTO (the number of shipped terminals deleted) with the **TRMSKDEL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC156 when the number of shipped terminals deleted exceeds the value specified by the **TRMSKDEL** guidance variable.

The default value for the **TRMSKDEL** guidance value is 50, indicating that Rule CIC156 will be produced when more than 50 shipped terminal definitions have been deleted during any CICS interval statistics period.

Suggestion: The CRMF transaction processing causes negligible CPU overhead. Consequently, there is little overhead incurred by specifying a low value for DSHIPINT (causing the CRMF transaction to be invoked relatively often).

CPExpert suggests that you consider specifying a relatively low value (for example, specify an hour or so) for the DSHIPINT variable. This relatively frequent execution of CRMF would result in redundant terminal definitions being identified as their idle time exceeds the DSHIPIDL value. These redundant terminal definitions would be deleted on a more regular basis and the CPU processing would be spread out over time, rather than the CPU overhead occurring in "bursts" of CPU overhead.

Reference: *CICS/ESA Version 4.1.1 Performance Guide*: Section 4.8.6 and Appendix A.1.1.

CICS/TS Release 1.1 Performance Guide: Section 4.8.7 and Appendix 1.1.2.

CICS/TS Release 1.2 Performance Guide: Section 4.8.7 and Appendix 1.1.2.

CICS/TS Release 1.3 Performance Guide: Section 4.12.7 and Appendix 1.1.2.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 24 (Deletion of shipped terminal definitions) and Appendix A (Table 17).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.11.7 (Controlling the deletion of shipped terminal definitions) and Appendix 1.1.1.2.

Rule CIC160: The number of strings may be too low for VSAM file

Finding: CPExpert has detected that I/O requests for VSAM files waited for VSAM strings.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built.

VSAM requires one or more strings for each concurrent file operation. One string will be required for each access if the file operation is directly to a base data set. Two strings will be required for each access if the file operation is through an AIX path (one string will be required to hold the position on the AIX and one string will be required to hold the position on the base data set).

An operation such as a read direct will free the string (or strings) immediately. However, a read for update, mass insert, or browse will retain the string (or strings) until a corresponding release, update, or end browse is performed.

For NSR VSAM files, the STRNO operand in the DFHFCT TYPE=DATASET macro (or the STRINGS operand in the Resource Definition Online FILE definition) specifies the number of concurrent requests to the data set. Performance is affected by an appropriate selection of the number of strings.

- If too few concurrent accesses are allowed, tasks may wait on strings. Unacceptable task response times may result from the waiting on strings.

- If more concurrent accesses are specified than required for response time performance, the excess strings will be unused. However, VSAM buffer space will be allocated for the defined strings. This space will increase virtual storage requirements and may cause CICS to have storage constraints.

The CICS statistics provide information about the number of times tasks were required to wait for strings. This information is provided for each VSAM file.

IBM's CICS Performance Guide indicates that, for NSR VSAM files, you should not try to eliminate all waits for strings since these may be a result of data set access patterns. For example, read operations may have to wait for the completion of browse operations on the same CI. Consequently, increasing the number of strings may have no positive effect on performance, but may simply require allocation of more virtual storage.

CPExpert divides the wait-on-string values in the CICS statistics by the total file access operations. CPExpert produces Rule CIC160 if the resulting percentage is greater than the STRWAIT guidance variable. The CICS Performance Guide indicates that it may be acceptable to have 5% of the file accesses waiting on strings for NSR files. Consequently, the default value for the STRWAIT guidance variable is 5%.

The percentage of file accesses waiting on strings clearly is a function of the file access patterns for the VSAM file. You may wish to change the STRWAIT guidance variable to a lower value to provide an alert about a **potential** constraint to improved performance.

Suggestion: CPExpert suggests that you consider increasing the value of the STRNO operand (or the STRINGS operand in the Resource Definition Online FILE definition) for the file indicated in this rule. You should then:

- Review the CICS statistics after operating with the new number of strings.
- If the wait on string for this file **has** decreased with the new value, continue operating with the new value.
- If the wait on string for this file has **not** decreased with the new value, the waits are a function of the access patterns of tasks referencing the file. In this case:
  - Return the value to its original setting. There is no point in allocating more strings to the file if the strings are not going to reduce the times tasks must wait on strings. The strings require virtual storage for buffers, and this storage is simply wasted.
  - Increase the value of the STRWAIT guidance variable in USOURCE(CICGUIDE) to eliminate the firing of Rule CIC160. There is no point in firing this rule if changing the STRNO value will not improve performance.

Note that before increasing the value of the STRWAIT guidance variable, you should have evaluated the effect of changing the STRNO operand for all files with high wait-on-string values.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 63-64 and pages 234-235.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 172, and pages 391-393.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 228-233.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-148 and page 289.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-159 and page 307.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.5 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.5 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.5 and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM string settings for NSR (STRINGS)) and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.5 (Defining VSAM string settings for NSR) and Appendix 1.1.11. |

Rule CIC161: The number of strings is too low for VSAM LSR pool

Finding: CPExpert has detected that I/O requests for files assigned to a Local Shared Resources (LSR) pool waited for VSAM strings.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built.

VSAM requires one or more strings for each concurrent file operation. One string will be required for each access if the file operation is directly to a base data set. Two strings will be required for each access if the file operation is through an AIX path (one string will be required to hold the position on the AIX and one string will be required to hold the position on the base data set).

An operation such as a read direct will free the string (or strings) immediately. However, a read for update, mass insert, or browse will retain the string (or strings) until a corresponding release, update, or end browse is performed.

The number of strings specified for the LSR pool restricts the total number of concurrent requests to the data sets assigned to the LSR pool. Additionally, the number of concurrent requests to individual data sets is restricted to the number of strings associated with each data set.

- If too few concurrent accesses are allowed, tasks may wait on strings. Unacceptable task response times may result from the waiting on strings.

- If more concurrent accesses are specified than required for response time performance, the excess strings will be unused. However, VSAM buffer space will be allocated for the defined strings. This space will increase virtual storage requirements and may cause CICS to have storage constraints.

For LSR VSAM files, the number of strings can be specified explicitly (using the STRNO operand DFHFCT TYPE=SHRCTL macro or using the STRINGS operand in the Resource Definition Online FILE definition).

Alternatively, CICS can automatically compute the number of strings for the LSR pool, based upon the characteristics of files assigned to the pool. There are advantages and disadvantages to allowing CICS to compute the number of strings in the LSR pool.

- It is easier to allow CICS to perform the computations. Additionally, if the number of strings in the LSR pool is explicitly defined, the definition usually should be altered when additional files are assigned to the LSR pool or when files are removed from the pool.

- However, allowing CICS to compute the number of strings for the LSR pool requires additional overhead (at startup) to build the LSR pool. This is because CICS must read the VSAM catalog for each file assigned to the LSR pool. Additionally, if CICS is allowed to compute the number of strings in the LSR pool, there is no ability to explicitly specify the number of strings for performance improvement.

It is generally better to explicitly specify the number of strings assigned to the LSR pool. The decreased flexibility of allowing CICS to compute the number of strings often outweighs the savings in programmer time required to make the specifications.

The CICS statistics provide information about the number of times tasks were required to wait for LSR pool strings. This information is provided for each LSR pool, rather than at the individual file level. (String wait information is provided at the file level only for waits caused by the number of strings assigned to each file. Information is not provided at the file level if the wait was caused by the number of strings specified for the LSR pool. However, information is provided at the file level for files waiting for LSR **buffers**.)

Tasks generally should not wait on strings for files assigned to a LSR pool. Consequently, CPExpert produces Rule CIC161 if the CICS statistics revealed that tasks waited on LSR pool strings.

Suggestion: CPExpert suggests that you consider the following:

- If you are explicitly defining the number of strings assigned to this LSR pool, increase the value of the STRNO operand (or the STRINGS operand in the Resource Definition Online FILE definition) for the LSR pool indicated in this rule.

- If you are allowing CICS to compute the number of strings assigned to this LSR pool, you should increase the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of strings required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that the maximum number of 255 strings is not violated and that the resulting value is at least as large as the largest number of strings specified for any file in the LSR pool.

  If you previously specified a value for the RSCLMT operand, the value should be increased. If you did not previously specify a value for the RSCLMT operand, specify a value larger than 50 for the operand. The value should be increased until the wait-on-string for the LSR pool is eliminated.

  CICS also uses the RSCLMT operand to control the number of buffers computed by CICS. Increasing the RSCLMT operand may result in an increase in the number of buffers, and a resulting increase in the virtual storage requirement. With Release 1.1 of DFP, the LSR buffers are allocated above the 16 megabyte line. Consequently, the increased requirement for virtual storage may not cause a virtual storage constraint.

  If CICS begins to experience a virtual storage constraint because of the increased buffers, you should cease allowing CICS to compute the strings and buffers for the LSR pool, and you should then explicitly specify the required strings and buffers using the SHRCTL macro.

  An additional consideration is that additional real storage is required for the increased strings and buffers. Increasing the number of strings (or buffers) may be inappropriate if CICS begins to experience increased paging delays.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68 and pages 232-238.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 173, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 93-106.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152 and page 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162 and page 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.6 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.6 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.6 and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM string settings for LSR (STRINGS)) and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.6 (Defining VSAM string settings for LSR) and Appendix 1.1.17.1.

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.6 (Defining VSAM string settings for LSR) and Appendix 1.1.17.1.

Rule CIC162: The number of strings may be too high for VSAM LSR pool

Finding: CPExpert has detected that storage was a constraint to CICS performance. However, a Local Shared Resources (LSR) pool had excessive strings allocated.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: The following rule causes this rule to be invoked:

Rule CIC110: CICS encountered a Short-on-Storage condition

Discussion: A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built.

VSAM requires one or more strings for each concurrent file operation. One string will be required for each access if the file operation is directly to a base data set. Two strings will be required for each access if the file operation is through an AIX path (one string will be required to hold the position on the AIX and one string will be required to hold the position on the base data set).

An operation such as a read direct will free the string (or strings) immediately. However, a read for update, mass insert, or browse will retain the string (or strings) until a corresponding release, update, or end browse is performed.

The number of strings specified for the LSR pool restricts the total number of concurrent requests to the data sets assigned to the LSR pool. Additionally, the number of concurrent requests to individual data sets is restricted to the number of strings associated with each data set.

- If too few concurrent accesses are allowed, tasks may wait on strings. Unacceptable task response times may result from the waiting on strings.

- If more concurrent accesses are specified than required for response time performance, the excess strings will be unused. However, VSAM buffer space will be allocated for the defined strings. This space will increase virtual storage requirements and may cause CICS to have storage constraints.

For LSR VSAM files, the number of strings can be specified explicitly (using the STRNO operand of the DFHFCT TYPE=SHRCTL macro or using the STRINGS operand in the Resource Definition Online FILE definition).

Alternatively, CICS can automatically compute the number of strings for the LSR pool, based upon the characteristics of files assigned to the pool. There are advantages and disadvantages to allowing CICS to compute the number of strings in the LSR pool.

- It is easier to allow CICS to perform the computations. Additionally, if the number of strings in the LSR pool is explicitly defined, the definition usually should be altered when additional files are assigned to the LSR pool or when files are removed from the pool.

- However, allowing CICS to compute the number of strings for the LSR pool requires additional overhead (at startup) to build the LSR pool. This is because CICS must read the VSAM catalog for each file assigned to the LSR pool. Additionally, if CICS is allowed to compute the number of strings in the LSR pool, there is no ability to explicitly specify the number of strings for performance improvement.

It is generally better to explicitly specify the number of strings assigned to the LSR pool. The decreased flexibility of allowing CICS to compute the number of strings often outweighs the savings in programmer time required to make the specifications.

The CICS statistics provide information about the number of strings that were concurrently used for each LSR pool. The number of unused strings is not normally a concern. However, CPExpert determined that storage was a constraint for the CICS region. Under this situation, any unused allocations of storage should be eliminated.

CPExpert produces Rule CIC162 if the CICS statistics revealed that storage was a constraint to improved CICS performance, and if less than 75% of the strings assigned to any LSR pool were concurrently used.

Suggestion: CPExpert suggests that you consider the following:

- If you are explicitly defining the number of strings assigned to this LSR pool, decrease the value of the STRNO operand (or the STRINGS operand in the Resource Definition Online FILE definition) for the LSR pool indicated in this rule.

  The number of strings specified for a LSR pool cannot be decreased to less than the maximum number of strings associated with any file assigned to the pool. If this restriction constrains you, perhaps you should examine the file definition to see if the number of strings is properly defined.

- If you are allowing CICS to compute the number of strings assigned to this LSR pool, you should decrease the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of strings required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that the maximum number of 255 strings is not violated and that the resulting value is at least as large as the largest number of strings specified for any file in the LSR pool.

  If you previously specified a value for the RSCLMT operand, the value should be decreased. If you did not previously specify a value for the RSCLMT operand, specify a value lower than 50 for the operand. The value should be decreased until the number of concurrently used strings for the LSR pool is slightly less than the specification. (Recall that the CICS statistics report the number of concurrently used strings for each LSR pool.)

  CICS also uses the RSCLMT operand to control the number of buffers computed by CICS. Decreasing the RSCLMT operand may result in a decrease in the number of buffers, and a resulting decrease in the virtual storage requirement. If CICS tasks begin to wait on buffers because of the decreased buffers, you should cease allowing CICS to compute the strings and buffers for the LSR pool, and you should then explicitly specify the required strings and buffers using the SHRCTL macro.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68 and pages 232-238.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 173, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 93-106.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152 and page 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162 and page 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.6 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.6 (Defining VSAM string settings for LSR), Section 4.5.4 (VSAM buffer allocations for LSR), and Appendix 1.1.17.1.

Rule CIC163: The number of buffers is too low for VSAM LSR pool

Finding: CPExpert has detected that file access requests to a Local Shared Resources (LSR) pool were required to wait for buffers. However, the CICS statistics did not reveal that storage was a constraint to CICS performance.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: A VSAM buffer is used to hold each VSAM control interval (CI) required to respond to a file access request. The number of data and index buffers are explicitly specified for Nonshared Resources (NSR) files. However, LSR files share a common pool of buffers, and there is no preallocation of buffers to particular files, to data or index, or to strings. Additionally, the LSR pool of buffers may be divided into "subpools" where each subpool consists of buffers of a specific size.

For LSR VSAM files, the number of buffers of each size can be specified explicitly (using the BUFFERS operand in the DFHFCT TYPE=SHRCTL macro or the DEFINE LSRPOOL command of Resource Definition Online).

Alternatively, CICS can automatically compute the number of buffers for the LSR pool, based upon the characteristics of files assigned to the pool. There are advantages and disadvantages to allowing CICS to compute the number of buffers in the LSR pool.

- It is easier to allow CICS to perform the computations. Additionally, if the number of buffers in the LSR pool is explicitly defined, the definition usually should be altered when additional files are assigned to the LSR pool or when files are removed from the pool.

- However, allowing CICS to compute the number of buffers for the LSR pool requires additional overhead (at startup) to build the LSR pool. This is because CICS must read the VSAM catalog for each file assigned to the LSR pool.

  Additionally, if CICS is allowed to compute the number of buffers in the LSR pool, there is no ability to explicitly specify the number of buffers for performance improvement.

It is generally better to explicitly specify the number of buffers assigned to the LSR pool. The decreased flexibility of allowing CICS to compute the number of buffers often outweighs the savings in programmer time required to make the specifications.

The CICS statistics provide information about the number of requests waiting for a buffer at any one time, and the total number of times request had to wait for a buffer. This information is provided by file name, and information is given about the size of the data and index buffers associated with the file.

Unfortunately, the information provided does not reveal whether the wait was for data or index buffers. Since the CI size is generally different for data and index, the information does not indicate which LSR subpool buffers were unavailable. However, CICS performance is constrained when any file access operations must wait for buffers.

CPExpert produces Rule CIC163 if the CICS statistics revealed that any files waited for buffers in an LSR pool. CPExpert provides information regarding the data buffer size and index buffer size associated with the file experiencing the wait for buffers.

Suggestion: CPExpert suggests that you consider the following:

- If you are explicitly defining the number of buffers assigned to this LSR pool, examine the DFHFCT TYPE=SHRCTL,BUFFER operand for the LSR pool. The BUFFER operand shows the CI size for the data and index buffers, and the number of buffers allocated to each size. You should increase the number of buffers for the size or sizes associated with the file.

  As mentioned above, the CICS statistics do not reveal which LSR subpool caused the waits. The wait on buffer could be for the LSR subpool of the data buffer size or for the LSR subpool of the index buffer size.

  However, the statistics do reveal the number of buffer reads and writes, by LSR subpool. CPExpert suggests that you increase the buffers of the LSR subpool with the largest number of buffer reads and writes. It is more likely that buffers of that size would be in short supply.

  The above conclusion is not certain, however. The wait on buffers could occur at relatively short intervals of high activity for a particular LSR subpool, even though the LSR subpool could have low **overall** activity.

Unfortunately, the approach must be to increase the buffer allocation for a LSR subpool and see whether that increase caused the wait on buffers problem to vanish. If the problem did not vanish, then the problem likely is related to the other LSR subpool.

- If you are allowing CICS to compute the number of buffers assigned to this LSR pool, you should increase the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of buffers of each size required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that there is at least three buffers in each LSR subpool).

  If you previously specified a value for the RSCLMT operand, the value should be increased. If you did not previously specify a value for the RSCLMT operand, specify a value higher than 50 for the operand. The value should be increased until there are no instances in which files wait on buffers.

  This method is imprecise and applies to all buffers in the LSR pool. The limitations of the method illustrate another advantage of explicitly specifying LSR pool operands, rather than allowing CICS to compute the operands.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68 and pages 232-238.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 173, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 93-106.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152 and page 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162 and page 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.1 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC164: The number of buffers may be too high for VSAM LSR pool

Finding: CPExpert has detected that storage was a constraint to CICS performance. However, a Local Shared Resources (LSR) pool potentially had excessive buffers allocated.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: The following rule causes this rule to be invoked:

Rule CIC110: CICS encountered a Short-on-Storage condition

Discussion: A VSAM buffer is used to hold each VSAM control interval (CI) required to respond to a file access request. The number of data and index buffers are explicitly specified for Nonshared Resources (NSR) files. However, LSR files share a common pool of buffers, and there is no preallocation of buffers to particular files, to data or index, or to strings. Additionally, the LSR pool of buffers may be divided into "subpools" where each subpool consists of buffers of a specific size.

For LSR VSAM files, the number of buffers of each size can be specified explicitly (using the BUFFERS operand in the DFHFCT TYPE=SHRCTL macro or the DEFINE LSRPOOL command of Resource Definition Online).

Alternatively, CICS can automatically compute the number of buffers for the LSR pool, based upon the characteristics of files assigned to the pool. There are advantages and disadvantages to allowing CICS to compute the number of buffers in the LSR pool.

- It is easier to allow CICS to perform the computations. Additionally, if the number of buffers in the LSR pool is explicitly defined, the definition usually should be altered when additional files are assigned to the LSR pool or when files are removed from the pool.

- However, allowing CICS to compute the number of buffers for the LSR pool requires additional overhead (at startup) to build the LSR pool. This is because CICS must read the VSAM catalog for each file assigned to the LSR pool. Additionally, if CICS is allowed to compute the number of buffers in the LSR pool, there is no ability to explicitly specify the number of buffers for performance improvement.

It generally is better to explicitly specify the number of buffers assigned to the LSR pool. The decreased flexibility of allowing CICS to compute the number of buffers often outweighs the savings in programmer time required to make the specifications.

The CICS statistics provide information about the number of requests waiting for a buffer at any one time, and the total number of times request had to wait for a buffer. This information is provided by file name, and information is given about the size of the data and index buffers associated with the file.

It is normally undesirable for VSAM I/O requests to wait for buffers in a LSR pool. However, the CICS statistics revealed that storage was a constraint to CICS performance. In this situation, CPExpert tries to identify possible ways in which storage can be conserved. VSAM buffers usually account for a significant amount of the real and virtual storage requirements of a CICS region. It may be prudent to allow some I/O access requests to wait for buffers if the alternative is unacceptable paging or a shortage of virtual storage.

CPExpert produces Rule CIC164 if the CICS statistics revealed that storage was a constraint to CICS performance, and if files assigned to the LSR pool **never** waited for buffers. CPExpert computes the total amount of virtual storage associated with the LSR pool and provides this information.

Suggestion: The suggestions associated with this rule should be followed only if (1) you believe that the number of buffers truly is excessive and (2) other opportunities to reduce the storage constraint have been unsuccessful.

This caution is presented because there normally are **significant** performance advantages in having a relatively large number of LSR buffers. A relatively large number of buffers usually implies a relatively large percentage of "read hits" (that is, I/O access operations are satisfied from the LSR buffers, rather than requiring a physical I/O operation to provide the data).

With the above caution in mind, CPExpert suggests that you consider the following:

- If you are explicitly defining the number of buffers assigned to this LSR pool, examine the DFHFCT TYPE=SHRCTL,BUFFER operand for the LSR pool. The BUFFER operand shows the CI size for the buffers, and the number of buffers allocated to each size. You should consider decreasing the number of buffers for associated with the LSR pool.

  In selecting a buffer size to reduce the number of buffers, it is generally advisable to reduce the buffers matching the data buffer size rather than the index buffer size. It normally is much more desireable to have read hits on index buffers than it is to have read hits on data buffers.

- If you are allowing CICS to compute the number of buffers assigned to this LSR pool, you should decrease the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of buffers of each size required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that there is at least three buffers in each LSR subpool).

  If you previously specified a value for the RSCLMT operand, the value should be decreased. If you did not previously specify a value for the RSCLMT operand, specify a value lower than 50 for the operand.

  This method is imprecise and applies to all buffers in the LSR pool. The limitations of the method illustrate another advantage of explicitly specifying LSR pool operands, rather than allowing CICS to compute the operands.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, page 165, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.1 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC165: The "look-aside" read hit ratio was low for VSAM LSR pool

Finding: CPExpert has detected that the "look-aside" read hit ratio was low for a VSAM Local Shared Resources (LSR) pool.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times. Some files will have requirements for buffers at one time, while at another time they will not be accessed and will not require buffers. The demand for buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

For example, 3 files might individually have a peak I/O access demand for 5 buffers. The sum of the individual buffers required to prevent buffer waits would be a total of 15 buffers (3 * 5). However, the peak collective demand would normally be less than 15 buffers. If there were no overlap of I/O access operations among the files, the peak collective demand would be only 5 buffers.

In practice, the peak collective demand for buffers is usually less than half of the sum of the peak individual demands. Assigning files to LSR pools therefore significantly decreases the storage requirements to support CICS VSAM buffers.

With CICS Version 1.7, the LSR buffers moved above the 16 megabyte line. Consequently, the storage savings generally are not as important as they were with previous versions of CICS.

However, there is an extremely important advantage to using LSR pools for VSAM files: VSAM will use its "look-aside" logic to determine whether a required control interval (CI) is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

With sufficient buffer allocation, VSAM often can find 80%-95% of the I/O requests for index records in buffers, and over 50% of the I/O requests for data records in buffers. Whether records are in a buffer is, of course, a function of the file size, the file accessing characteristics, etc. However, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS.

The CICS statistics provide information about the number of times an I/O request was satisfied because VSAM found the data in a LSR buffer. This value is titled "LOOK-ASIDE HITS" in the statistics. The CICS statistics also provide information about the number of times an I/O request was NOT satisfied by data in a LSR buffer. This value is titled "BUFFER READS" in the statistics. These two values are provided for each LSR pool and LSR subpool (a LSR subpool refers to the buffers of a particular size in the LSR pool).

CPExpert analyzes statistics relating to the files assigned to LSR pools to determine the subpool sizes used for data and index buffers. The LSR pool "look-aside hits" statistics are then evaluated based upon whether the subpools are used for data or index records.

CPExpert calculates the percent of "look-aside hits" versus total I/O access operations (the sum of "look-aside hits" and "buffer reads"). The resulting percent is compared with one of two guidance variables, depending upon how the subpool is used. If the subpool is used for data records, the percent is compared to the LSRHITD guidance variable. If the subpool is used for index records, the percent is compared to the LSRHITI guidance variable. The default values for the LSRHITD and LSRHITI guidance variables are 40% and 80%, respectively.

If the subpool is used for both data and index records, CPExpert tests the index LSRHITI guidance variable first. This is done since index records should have a higher hit ratio than data records.

CPExpert produces Rule CIC165 if the percent of "look-aside hits" is lower than the appropriate guidance variable.

Suggestion: As mentioned above, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS. Under many circumstances, Rule CIC165 should mean that additional buffers should be allocated to the LSR subpool.

The normal technique is to add buffers to the LSR subpool until the "look-aside hit" read ratio is at an acceptable level. This approach must be guided by the availability of virtual storage. Further, there usually is no point in adding buffers once the "look-aside hit" read ratio ceases increasing.

There are a number of unique situations, however, in which allocating more buffers to the LSR subpool may not be an appropriate action.

- The files are very large and their processing characteristics imply that there is little probability that a required record will be in a LSR buffer.
- A large active file is processed in a manner such that it dominates the buffer subpool. In this case, you may wish to place the file into a LSR pool of its own.
- Storage is a constraint to CICS performance (or to overall system performance). Increasing the number of LSR buffer pools may be unacceptable in this situation.
- Data records and index records have the same CI size for some files assigned to the LSR subpool. In this situation, the data records often will dominate the subpool and there will be a low "look-aside hit" ratio for the subpool.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC166: VSAM LSR files have same CI size for data and index records

Finding: CPExpert has detected that VSAM files assigned to a Local Shared Resources (LSR) pool have been defined with the same control interval (CI) size for data and index records.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Please refer to Rule CIC165 for more discussion of this issue.

The size of the CIs for data sets is not an operand specified to CICS. Rather, the size of the CIs is specified using VSAM (using the IDCAMS utility). It is common to specify the CI for data records, and to allow VSAM to select an appropriate CI for index records. (In fact, VSAM will override any specified CI size for index records if VSAM decides that the specified size is insufficient.)

LSR pools are generated using the Resource Definition Online (RDO) LSRPOOL definition, or using the DFHFCT TYPE=SHRCTL macro. The BUFFERS operand of this macro is used to define the number and size of the buffers assigned to the pool. Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the CI for data and index records of files assigned to the LSR pool. The pools with different buffer sizes are referred to as LSR subpools.

When executing an I/O operation, VSAM selects a buffer sufficiently large to hold the CI, regardless of whether the CI is for data or index records. Buffers for index CIs and data CIs will be drawn from the same buffer subpool if any file has index records with the same CI as the data records for any file in the LSR pool. CICS performance can be severely degraded if any LSR subpool services both data records and index records. This is because the VSAM "look-aside" logic will tend to find CIs for data records in the buffers, rather than finding CIs for index records.

A major advantage of using LSR pools for VSAM files is that VSAM will use its "look-aside" logic to determine whether a required CI is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

With sufficient buffer allocation, VSAM often can find 80%-95% of the I/O requests for index records in buffers, and over 50% of the I/O requests for data records in buffers. Whether records are in a buffer is, of course, a function of the file size, the file accessing characteristics, etc. However, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS.

VSAM uses a least-recently-used (LRU) algorithm in deciding when to reuse a buffer in a subpool. There normally are many more CIs for data records than there are CIs for index records. Consequently, the CIs for data records will dominate a subpool if the CIs are the same for data and index records. This effect is exactly the opposite of what is desired for performance, and significantly degrades the VSAM "look-aside" logic.

The CI size of the data and index records is provided for each file in the CICS statistics. CPExpert analyzes this information to determine whether there are any conflicts between CI size for data and index records for any files assigned to LSR subpools. CPExpert produces Rule CIC166 if any LSR subpool services VSAM files where any data CI is the same size as any index CI.

CPExpert makes sure that the files have a reasonably large number of I/O operations before firing Rule CIC166 (there is little point in worrying about this situation if the file has few accesses). CPExpert arbitrarily selected 1,000 as the threshold of file accesses; CPExpert will not be produced Rule CIC166 if the number of buffer reads (this means physical I/O operations) is less than 1,000 for any LSR subpool having a conflict with the CI size of data or index records.

Additionally, CPExpert examines the "look-aside" read hit ratio (described in Rule CIC165). CPExpert does not produce Rule CIC166 if this ratio is greater than 90% for the VSAM file(s) involved. (CPExpert concludes that there is little harm being done by the same size specification, and does not produce the rule spuriously.)

Note that CICS/ESA provides separate LSR buffer pools for data and index records. However, if only data buffers are specified for the LSR pool, only one set of buffers are built. This set of buffers will be used for both data and index records. Consequently, this rule applies to CICS/ESA if only data buffers were specified for the LSR pool.

The initial version of the CICS Component does not examine the DFHFCT macros to determine specifications. Consequently, Rule CIC166 may be produced in CICS/ESA environments when there is no real problem (that is, you may have specified separate LSR buffer pools for data and index records in CICS/ESA). Depending upon user feedback, future versions of the CICS Component may analyze the DFHFCT macros.

Suggestion: Allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS. This improvement occurs mostly because of the VSAM "look-aside" logic. However, the "look-aside" logic will normally function poorly if data and index files assigned to the LSR pool have the same CIs.

CPExpert suggests that you consider the following alternatives:

- Analyze the data and index CI sizes for the files identified by Rule CIC166. Determine whether the data or index CI sizes can be changed so that VSAM will use buffers from different LSR buffer subpools.

- If the VSAM file characteristics cannot be changed, consider assigning the conflicting files to different LSR pools. This option eliminates the advantage of minimizing buffer requirements among the files. However, this advantage is not as significant with CICS Version 1.7, since the buffer allocation is above the 16 megabyte line. This option may allow the VSAM "look-aside" logic to significantly improve CICS performance by eliminating many physical I/O operations associated with the file.

- If neither of the above options are feasible, consider assigning more buffers to the LSR buffer subpool. This option may not improve performance, however, depending upon the file accessing characteristics.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC167: There may be too few VSAM files assigned to LSR Pools

Finding: CPExpert has detected that there may be too few CICS VSAM files assigned to a Local Shared Resources (LSR) pool.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files can be assigned as nonshared resources (NSR) files or assigned to a local shared resources (LSR) pool. The major difference between the two methods is how the VSAM strings and VSAM buffers are allocated and used.

When CICS VSAM files are assigned as NSR files, there is no sharing of VSAM strings or VSAM buffers among files.

- Strings are assigned to files using the STRNO operand of the DFHFCT TYPE=FILE macro. The number of assigned strings controls the number of concurrent I/O requests to the file.

- Buffers are assigned to files using the BUFND operand (for data buffers) and BUFNI operand (for index buffers) of the DFHFCT TYPE=FILE macro. The number of assigned buffers for data records and for index records must be at least one more than the number of strings. Additional data buffers may be assigned (usually this is done to speed up VSAM CA splits by permitting chained I/O operations). Additional index buffers may be assigned and will be shared between the strings (the extra buffers can be used to hold high-level index records and thus save physical I/O).

When CICS VSAM files are assigned to LSR pools, the files share common strings and buffers assigned to the LSR pool. There is no "preallocation" of strings or buffers to particular files. Since the strings and buffers are shared, significantly fewer strings and buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Please refer to Rule CIC165 for more discussion of this issue.

- Strings are assigned to LSR pools using the Resource Definition Online (RDO) **LSRPOOL** definition, or using the STRNO operand of the DFHFCT TYPE=SHRCTL macro. Alternatively, the number of strings assigned to the LSR pool can be automatically computed by CICS using an algorithm based on the number of strings associated with files assigned to the pool. The number of assigned strings controls the number of concurrent I/O requests to the LSR pool. This number is normally much less than the sum of the strings associated with files assigned to the LSR pool (The default computed value is 50% of the number of strings assigned to files, with adjustments to ensure that the minimum required strings are assigned.).

- Buffers are assigned to LSR pools using the BUFFERS operand of the DFHFCT TYPE=SHRCTL macro. Alternatively, the number of buffers assigned to the LSR pool can be automatically computed by CICS using an algorithm based on the number of strings associated with files assigned to the pool. This number is normally much less than the sum of the strings associated with files assigned to the LSR pool (The default computed value is 50% of the number of strings assigned to files, with adjustments to ensure that the minimum required buffers are assigned.). Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the control interval (CI) for data and index records of files assigned to the LSR pool.

There are some advantages to assigning files as NSR files:

- NSR allows multiple copies of a control interval (CI) in storage. An I/O request associated with one string can be updating a CI, while I/O requests associated with other strings are reading different copies of the same CI. This "read while update" is generally considered to be a poor practice, and many installations implement standards to prevent such situations.

- NSR provides better performance for VSAM CA splits if extra data buffers are allocated to the file.

- NSR allows allocation of buffers to a specific file. This advantage may be important in some unique situations with critical files. However, this advantage is normally not applicable, since critical files can be assigned to their own LSR pool and benefit from the improved "look-aside" logic implemented with LSR pools (see the discussion below).

- NSR can sometimes provide slightly better performance for sequential operations if additional buffers are allocated. This is because the processing overhead required to implement the "look-aside" logic implemented with LSR pools is not required.

There are major advantages to assigning VSAM files to LSR pools:

- If VSAM files are assigned to LSR pools, VSAM will use its "look-aside" logic to determine whether a required CI is already in a buffer, before executing any physical I/O operations. If the required record is already in a buffer, VSAM will use the record in storage, rather than issuing a read to DASD. This has the effect of implementing an in-storage caching of the file, and can **significantly** reduce the number of physical I/O operations required.

  With sufficient buffer allocation, VSAM often can find 80%-95% of the I/O requests for index records in buffers, and over 50% of the I/O requests for data records in buffers. Whether records are in a buffer is, of course, a function of the file size, the file accessing characteristics, etc. However, allocating sufficient buffers to LSR pools (and assigning VSAM files to the LSR pools) often can produce a **significant** performance improvement for CICS.

- VSAM files assigned to a LSR pool share common strings and common buffers assigned to the LSR pool. Since the strings and buffers are shared, significantly fewer strings and buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times.

  Some files will have requirements for strings and buffers at one time, while at another time they will not be accessed and will not require the strings or buffers. The demand for strings and buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

  For example, 3 files might individually have a peak I/O access demand for 5 buffers. The sum of the individual buffers required to prevent buffer waits would be a total of 15 buffers (3 * 5). However, the peak collective demand would normally be less than 15 buffers. If there were no overlap of I/O access operations among the files, the peak collective demand would be only 5 buffers.

  In practice, the peak collective demand for buffers is usually less than half of the sum of the peak individual demands. Assigning files to LSR pools therefore significantly decreases the storage requirements to support CICS VSAM buffers.

  With CICS Version 1.7, the LSR buffers moved above the 16 megabyte line. Consequently, the storage savings generally are not as important as they were with previous versions of CICS.

- Assigning files to LSR pools provides better read integrity, since there is only one copy of a CI in storage. LSR permits several read operations to share access to the same buffer. However, updates require the exclusive use of the buffer. No other I/O request (e.g., a read or another update) is allowed access to the buffer until the update releases the buffer.

CPExpert analyzes the number of I/O requests for LSR files and computes this as a percent of all I/O requests for all VSAM files. CPExpert produces Rule CIC167 if this percent is less than the LSRIOREQ guidance variable.

The default of the LSRIOREQ guidance variable is 75, indicating that 75% of the I/O requests should be satisfied from files assigned to LSR pools. This default is set such that Rule CIC167 probably will initially be produced for many CICS regions. Many installations have allocated a relatively small number of files to LSR pools. The point of firing Rule CIC167 is to alert you to the significant performance advantages of using LSR pools. If you are not inclined to assign more VSAM files to LSR pools, change the guidance variable and prevent the firing of this rule.

Suggestion: CPExpert suggests that you assign more VSAM files to LSR pools. **The advantages of using LSR pools are so significant that VSAM files generally should be assigned to LSR pools.** There are some exceptions to this general statement:

- The file undergoes frequent VSAM control area (CA) splits.
- Files might be assigned as NSR files if there is no opportunity for the LSR "look-aside" logic to provide performance benefits. This situation could arise if the file is very active, and is a large sequentially-processed files.
- High performance is required for a critical file and the processing is unique in some way that would benefit from NSR. This exception is normally not applicable, since critical files can be assigned to their own LSR pool and benefit from the improved "look-aside" logic implemented with LSR pools.
- The processing characteristics of a particular file would dominate a LSR pool. In this case, the file probably would benefit more by assigning it to its own LSR pool, rather than assigning it as a NSR file.

However, unless you have a unique situation, you should normally assign most of your CICS VSAM files to LSR pools.

Additionally, you may wish to review the discussion in Rule CIC160 through Rule CIC166, and Rule CIC168. These rules provide insight into some of the implications of various implementation strategies.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC168: LSR buffer sizes may be inappropriate

Finding: CPExpert has detected that the data buffer size or index buffer size of VSAM files assigned to a Local Shared Resources (LSR) pool do not match the size of the LSR buffers.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM files can be assigned as nonshared resources (NSR) files or assigned to a local shared resources (LSR) pool. The major difference between the two methods is how the VSAM strings and VSAM buffers are allocated and used. Please refer to Rule CIC167 for further explanation of this difference.

When CICS VSAM files are assigned to LSR pools, the files share common strings and buffers assigned to the LSR pool. There is no "preallocation" of strings or buffers to particular files. Since the strings and buffers are shared, significantly fewer strings and buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time.

Buffers are assigned to LSR pools using the BUFFERS operand of the DFHFCT TYPE=SHRCTL macro. Alternatively, the number of buffers assigned to the LSR pool can be automatically computed by CICS using an algorithm based on the number of strings associated with files assigned to the pool. This number is normally much less than the sum of the strings associated with files assigned to the LSR pool (The default computed value is 50% of the number of strings assigned to files, with adjustments to ensure that the minimum required buffers are assigned.).

Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the control interval (CI) for data and index records of files assigned to the LSR pool. The buffer size specification can be made using the BUFFERS operand (as described above).

Alternatively, the buffer size can be automatically computed by CICS, when CICS is computing the number of buffers in the LSR pool. CICS computes the buffer size based upon the requirements of the data and index record sizes of the files assigned to the LSR pool.

If CICS computes the buffer sizes, the LSR pool will have buffers assigned to match the size of the files assigned to the LSR pool. However, if the buffer sizes are defined using the BUFFERS operand, the user has the responsibility of assigning the number of buffers and the buffer sizes. In this case, the buffer sizes may not match the size of the data or index records of the files assigned to the LSR pool.

If there are no buffers with a size matching the data or index CI sizes of a file, VSAM will select the next largest buffer size that is assigned to the LSR pool. For example, suppose that the LSR pool were assigned buffers of 2048 bytes and 4096 bytes. If a data or index CI size for a file were 1024 bytes, VSAM would select the next largest size buffer (in this example, VSAM would select buffers from the 2048 byte pool).

This "mismatch" of CI sizes to buffer pool sizes has two effects:

- Buffer space is wasted, since a data or index record of 1024 will require a buffer of 2048. This results in an inefficient use of storage.

- An unnecessarily large number of buffers may be specified for the larger buffer pool, to prevent files waiting on buffers.

CPExpert analyzes the size of the data and index CIs for files assigned to each LSR buffer pool. CPExpert produces Rule CIC168 if there are any data or index CIs without corresponding buffer sizes assigned to the LSR pool.

Suggestion: CPExpert suggests that you assign VSAM buffers to the LSR pool with sizes corresponding to the size of the data or index CIs of the files assigned to the LSR pool.

An initial reaction to this type of problem might be to "Let CICS compute the buffer size." However, there are many situations in which CICS will not compute an appropriate number of buffers or strings for a LSR pool. (Other rules in the CIC16x series give examples of inappropriate computation of the number of buffers or buffer strings.)

So long as storage is **not** a constraint for the CICS region, CICS can satisfactorily compute the number of buffers and strings. CICS can be directed to increase the number of buffers and strings by increasing the value of the RSCLMT operand in the SIT. If storage is not a constraint, this may be an acceptable approach, since CICS will adjust the number of buffers and strings based upon the value of the RSCLMT operand.

However, if storage **is** a constraint, allowing CICS to compute the number of buffers may result in an inappropriate number of buffers and strings.

- CICS computes the required number of buffers of each size based upon the STRNO operand in the DFHFCT entry, the number of levels in the index for index components, etc. After the total number of buffers of each required size is computed, the number of buffers in each size is reduced to either 50% of the computed value or to the percentage specified in the RSCLMT operand (a minimum of 3 buffers and a maximum of 255 strings are imposed on the result).

  For example, if the RSCLMT operand specified 65%, the number of buffers and strings would be reduced to 65% of the computed total for each size. If the RSCLMT operand specified 35%, the number of buffers and strings would be reduced to 35% of the computed total for each size.

- The RSCLMT operand applies to all buffer sizes and to all LSR pools. Consequently, it is not possible to adjust the number of buffers based upon the requirements of the files assigned to individual LSR pools or based upon the buffer sizes generated for each LSR pool.

- A significant amount of storage may be wasted if CICS is allowed to compute the buffer and string values associated with LSR pools. This is especially true if the RSCLMT operand is increased to a value necessary to improve the buffer "look-aside" read hit ratio for specific buffer sizes or to eliminate the wait-on-buffer condition of specific files.

If storage is a constraint, CPExpert suggests that you use the BUFFERS operand to explicitly define the buffer sizes and number of buffers in LSR pools for the CICS region. The buffer sizes should match the data and index CI sizes of the files assigned to the LSR pool.

There can be an unusual situation in which you may wish to not match all buffer sizes with the data and index CI sizes of the files assigned to the LSR pool. Suppose you have a file with data or index CI sizes that do not match the sizes of the other files, and further suppose that the file has a very low activity rate. If you specify buffers of this size for the LSR pool, the virtual storage will be reserved for the buffers even though the file rarely uses the buffers. When the file is not using the buffers, the virtual storage is wasted. Of course, this does not matter if virtual storage is not a constraint. However, if virtual storage is a constraint, you may wish to deliberately not assign buffers of the size necessary to accommodate the file, and allow the file to default to the next larger buffer size on the rare times when it is accessed.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Rule CIC169: The number of buffers may be too low for VSAM LSR pool

Finding: CPExpert has detected that a Local Shared Resources (LSR) pool experienced non-user initiated buffer writes. Non-user initiated buffer writes normally mean that an insufficient number of buffers are assigned to the LSR subpool.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: **User** initiated buffer writes are the writing of buffers due to a direct request from the user program. For example, EXEC CICS WRITE of a single record causes the record to be written to the file immediately.

Non-user initiated buffer writes are usually encountered only when deferred requests are involved. For example, if a program issues: EXEC CICS WRITE MASSINSERT, the first request starts a sequential operation. The sequential operation defers writing the control intervals (CIs) until an UNLOCK is issued.

Suppose the program were inserting 100 records and there is space for only 10 records in each CI. The 100 records would fill 10 CIs with data when the UNLOCK is issued. At that time, all 10 CIs (buffers) would be written. This would be a user initiated buffer write.

However, suppose other CICS transactions were also using buffers in the same LSR subpool while the transaction doing the MASSINSERT was running and while it was filling buffers. Suppose one of these transaction issued a VSAM request which needs a buffer, but there were no free buffers in the pool because of the MASSINSERT. Under this condition, one of the buffers that was filled by the MASSINSERT (but not yet written), may be forced out by VSAM's buffer replenishment routine. This is the non-user initiated buffer write.

A non-user initiated buffer write could also happen (using the MASSINSERT example given above) if less than 10 buffers were defined in the LSR subpool. The task doing the MASSINSERT would WRITE 100 records requiring 10 buffers and could force out some of its own buffers.

Of course, the 100 records and 10 buffers were used only as an example. Any other combination of operation/buffers could cause non-user initiated buffer writes if there were insufficient buffers allocated to the LSR subpool

CPExpert produces Rule CIC169 if the CICS statistics revealed that the number of non-user initiated buffer writes for any LSR subpool was greater than the **NONUSRBF** guidance variable. The default value for the NONUSRBF guidance variable is zero. CPExpert provides information regarding the subpool size associated with the non-user initiated buffer writes.

Suggestion: CPExpert suggests that you consider adding more buffers to the LSR subpool experiencing the non-user initiated buffer writes.

- If you are explicitly defining the number of buffers assigned to this LSR pool, examine the DFHFCT TYPE=SHRCTL,BUFFER operand for the LSR pool. The BUFFER operand shows the CI size for the data and index buffers, and the number of buffers allocated to each size. You should increase the number of buffers for the size or sizes associated with the file.

- If you are allowing CICS to compute the number of buffers assigned to this LSR pool, you should increase the value of the RSCLMT operand in the SHRCTL macro.

  After CICS computes the total number of buffers of each size required by all files assigned to the LSR pool, CICS reduces this number by 50% or to the percentage specified in the RSCLMT operand (the RSCLMT operand value takes precedence). CICS makes sure that there is at least three buffers in each LSR subpool).

  If you previously specified a value for the RSCLMT operand, the value should be increased. If you did not previously specify a value for the RSCLMT operand, specify a value higher than 50 for the operand. The value should be increased until there are no instances in which files wait on buffers.

  This method is imprecise and applies to all buffers in the LSR pool. The limitations of the method illustrate another advantage of explicitly specifying LSR pool operands, rather than allowing CICS to compute the operands.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/MVS Version 2.1.2 Performance* Guide: pages 158-162, page 173, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73 and pages 93-106.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152 and page 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162 and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2 and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2 and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

IBMLINK Document ID Q364609
TITLE: NON-USER INITIATED BUFFER WRITES AND USER INITIATED BUFFER WRITES

Rule CIC170: More than one string specified for write-only ESDS file

Finding: CPExpert has detected that a VSAM ESDS was used exclusively for write operations and that the ESDS had more than one string specified.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region. The level of impact depends on the amount of write activity that occurred and the number of strings concurrently used.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: An entry-sequenced data set is one in which each record is identified by its relative byte address (RBA).

Records are held in an ESDS in the order in which they were first loaded into the data set. New records that are added to an ESDS always go after the last record in the data set. Records in an ESDS may not be deleted, nor can the record lengths be altered. After a record has been stored in an ESDS, its relative byte address (RBA) remains constant. When browsing, records are retrieved in the order in which they were added to the data set.

A record added to an ESDS is always added to the end of the file. Records cannot be inserted in an ESDS between existing records. After the operation is completed, the relative byte address in the file where the record was placed is returned to the application program.

A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built. The number of concurrent requests to individual data sets is restricted to the number of strings associated with each data set.

The number of concurrent requests to the data set is specified by (1) the STRNO operand in the DFHFCT TYPE=DATASET macro (for NSR VSAM files), (2) the STRNO operand DFHFCT TYPE=SHRCTL macro (for LSR VSAM files), or (3) the STRINGS operand in the Resource Definition Online FILE definition. Alternatively, CICS can automatically compute the number of strings for the LSR pool for LSR VSAM files, based upon the characteristics of files assigned to the pool.

If no string is available when a file access is attempted, the file access must for an available string. Some waiting is tolerable for NSR VSAM files, since the number of strings concurrently required is highly dependent on the file access patterns. Tasks generally should not wait on strings for files assigned to a LSR pool for LSR VSAM files. Rule CIC160 and Rule CIC161 analyze waits for available strings for NSR VSAM files and LSR pools, respectively.

There are some special performance considerations when choosing a STRINGS valus for an ESDS file.

If an ESDS VSAM file is used as an 'add-only' file (that is, it is used only in write mode to add records to the end of the file), IBM strongly recommends that a single string be specified. Any string number greater than one can significantly affect performance, because of exclusive control conflicts that occur when more than one task attempts to write to the ESDS at the same time. With a string number greater than one, the cost of resolving exclusive control conflicts is greater than waiting for a string. Each time exclusive control is returned, a GETMAIN is issued for a message area, followed by a second call to VSAM to obtain the owner of the control interval.

If an ESDS VSAM file is used for both writing and reading, with writing being 80% of the activity, IBM suggesets that it is better to define two file definitions--using one file for writing and the other for reading.

CPExpert analyzes all files in which the A17DSTYP variable in CICFCT data set indicates that the file is VSAM ESDS data set type, and the A17STRNO value is greater than one. CPExpert examines the file activity characteristics for the ESDS to determine the read/write characteristics. CPExpert produces Rule CIC170 if the CICS statistics revealed that the file activity is exclusively write operations.

This finding does not apply to ESDS participating in record level sharing (RLS). For RLS, the value specified for strings is ignored and a value of 1024 strings is returned after OPEN of the ESDS (indicating the maximum number of strings). Consequently, CPExpert tests A17DSRLS in CICFCR to verify that the ESDS is a non-RLS file.

Suggestion: CPExpert suggests that you consider specifying only one string in the ESDS file definition.

Before taking this action, you should verify that the file access characteristics in the CICS interval statistics analyzed by CPExpert are representative of the ESDS (that is, verify that the ESDS is used exclusively for write activity during normal CICS processing).

Reference: *CICS/ESA Version 4.1.1 Performance* Guide: Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.1.2.1 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.1 and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM and file control - number of strings considerations for ESDS files) and Appendix A (Table 61).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.1.2.1 (Number of strings considerations for ESDS files) and Appendix 1.1.11.

Rule CIC171: Significant percent of write activity for ESDS file

Finding: CPExpert has detected that write operations accounted for a significant percent of the file activity to a VSAM ESDS.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region. The level of impact depends on the amount of write activity that occurred, the number of other file activity that occurred, and the number of strings concurrently used.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: An entry-sequenced data set is one in which each record is identified by its relative byte address (RBA).

Records are held in an ESDS in the order in which they were first loaded into the data set. New records that are added to an ESDS always go after the last record in the data set. Records in an ESDS may not be deleted, nor can the record lengths be altered. After a record has been stored in an ESDS, its relative byte address (RBA) remains constant. When browsing, records are retrieved in the order in which they were added to the data set.

A record added to an ESDS is always added to the end of the file. Records cannot be inserted in an ESDS between existing records. After the operation is completed, the relative byte address in the file where the record was placed is returned to the application program.

A VSAM "string" is a request to a VSAM data set requiring "positioning" within the data set. Each string results in a number of VSAM control blocks being built. The number of concurrent requests to individual data sets is restricted to the number of strings associated with each data set.

The number of concurrent requests to the data set is specified by (1) the STRNO operand in the DFHFCT TYPE=DATASET macro (for NSR VSAM files), (2) the STRNO operand DFHFCT TYPE=SHRCTL macro (for LSR VSAM files), or (3) the STRINGS operand in the Resource Definition Online FILE definition. Alternatively, CICS can automatically compute the number of strings for the LSR pool for LSR VSAM files, based upon the characteristics of files assigned to the pool.

If no string is available when a file access is attempted, the file access must for an available string. Some waiting is tolerable for NSR VSAM files, since the number of strings concurrently required is highly dependent on the file access patterns. Tasks generally should not wait on strings for files assigned to a LSR pool for LSR VSAM files. Rule CIC160 and Rule CIC161 analyze waits for available strings for NSR VSAM files and LSR pools, respectively.

There are some special performance considerations when choosing a STRINGS value for an ESDS file.

If an ESDS VSAM file is used as an 'add-only' file (that is, it is used only in write mode to add records to the end of the file), IBM strongly recommends that a single string be specified. Any string number greater than 1 can significantly affect performance, because of exclusive control conflicts that occur when more than one task attempts to write to the ESDS at the same time. With a string number greater than one, the cost of resolving exclusive control conflicts is greater than waiting for a string. Each time exclusive control is returned, a GETMAIN is issued for a message area, followed by a second call to VSAM to obtain the owner of the control interval.

If an ESDS VSAM file is used for both writing and reading, with writing being 80% of the activity, IBM suggests that it is better to define two file definitions--using one file for writing and the other for reading.

CPExpert analyzes all files in which the A17DSTYP variable in CICFCR data set indicates that the file is VSAM ESDS data set type and the A17STRNO value is greater than one. CPExpert examines the file activity characteristics for the selected ESDS to determine the read/write characteristics. CPExpert produces Rule CIC171 if the CICS statistics revealed that write operations accounted for more than 80% of the file activity

This finding does not apply to ESDS participating in record level sharing (RLS). For RLS, the value specified for strings is ignored and a value of 1024 strings is returned after OPEN of the ESDS (indicating the maximum number of strings). Consequently, CPExpert tests A17DSRLS in CICFCR to verify that the ESDS is a non-RLS file.

Suggestion: CPExpert suggests that you consider specifying two file definitions for the ESDS.

Before taking this action, you should verify that the file access characteristics in the CICS interval statistics analyzed by CPExpert are representative of the ESDS during normal CICS processing.

Reference: *CICS/ESA Version 4.1.1 Performance* Guide: Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.1.2.1 and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.1 and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM and file control - number of strings considerations for ESDS files) and Appendix A (Table 61).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.1.2.1 (Number of strings considerations for ESDS files) and Appendix 1.1.11. |

Rule CIC175: The VSAM LSR pool was seldom used

Finding: CPExpert has detected that a VSAM Local Shared Resources (LSR) pool was seldom used.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. This should be ignored unless storage is a constraint to CICS performance.

Discussion: VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times. Some files will have requirements for buffers at one time, while at another time they will not be accessed and will not require buffers. The demand for buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

For example, 3 files might individually have a peak I/O access demand for 5 buffers. The sum of the individual buffers required to prevent buffer waits would be a total of 15 buffers (3 * 5). However, the peak collective demand would normally be less than 15 buffers. If there were no overlap of I/O access operations among the files, the peak collective demand would be only 5 buffers.

In practice, the peak collective demand for buffers is usually less than half of the sum of the peak individual demands. Assigning files to LSR pools therefore significantly decreases the storage requirements to support CICS VSAM buffers.

Please refer to Rule CIC165 for further discussion of the benefits of using LSR pools.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the pools. If VSAM files do not use the LSR pools that are created, the storage dedicated to a particular pool might be better used elsewhere in CICS.

The CICS statistics provide information about the number of times an I/O request was made to a LSR pool.

CPExpert calculates the total I/O requests made to each LSR pool that was defined. The resulting value is compared with the LSRUSE guidance variable. CPExpert produces Rule CIC175 if the total number of I/O requests to the LSR pool is lower than the LSRUSE guidance variable. The default values for the LSRUSE guidance variable is 100, meaning that CPExpert will produce Rule CIC175 when less than 100 I/O requests are made to the LSR pool.

CPExpert will not produce Rule CIC175 unless the condition exists for more than 50% of the CICS statistics intervals being analyzed.

Suggestion: As mentioned above, if VSAM files do not use the LSR pools that are created, the storage dedicated to a particular pool might be better used elsewhere in CICS.

- If storage is a constraint to CICS performance, and Rule CIC175 is produced for a particular LSR pool, you should consider eliminating the LSR pool so the storage can be used elsewhere in CICS. You should, of course, identify any VSAM files that have specified the LSR pool and change their specification to a different LSR pool.

- You can alter CPExpert's analysis by specifying a different value for the LSRUSE guidance variable.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Thanks: Thanks to Paul Gordon (Bank of America) for suggesting this rule.

Rule CIC176: The VSAM LSR subpool buffers were seldom used

Finding: CPExpert has detected that VSAM Local Shared Resources (LSR) subpool buffers were seldom used.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. This should be ignored unless storage is a constraint to CICS performance.

Discussion: VSAM files assigned to a LSR pool share common buffers (and also share strings) assigned to the LSR pool. Since the buffers are shared, significantly fewer buffers normally are required to support I/O access operations. This is because not all files will be accessed at any particular time. Rather, file accesses will tend to be distributed across files at different times. Some files will have requirements for buffers at one time, while at another time they will not be accessed and will not require buffers. The demand for buffers therefore is the **peak collective demand** rather than the **sum** of the **peak individual** demands.

For example, 3 files might individually have a peak I/O access demand for 5 buffers. The sum of the individual buffers required to prevent buffer waits would be a total of 15 buffers (3 * 5). However, the peak collective demand would normally be less than 15 buffers. If there were no overlap of I/O access operations among the files, the peak collective demand would be only 5 buffers.

In practice, the peak collective demand for buffers is usually less than half of the sum of the peak individual demands. Assigning files to LSR pools therefore significantly decreases the storage requirements to support CICS VSAM buffers.

LSR pools are generated using the DEFINE LSRPOOL command of Resource Definition Online, or using the DFHFCT TYPE=SHRCTL macro. The BUFFERS operand of this macro is used to define the number and size of the buffers assigned to the pool. Buffers are normally specified with different sizes, with the different sizes corresponding to the size of the CI for data and index records of files assigned to the LSR pool. The pools with different buffer sizes are referred to as LSR subpools. There can be (and should be) different LSR subpools for data and index CI sizes[1]

Please refer to Rule CIC165 for further discussion of the benefits of using LSR pools.

[1] See Rule CIC166 for further discussion of this issue.

The discussion regarding benefits of using LSR pools assumes that VSAM files actually use the subpools. If VSAM files do not use the LSR subpools that are created, the storage dedicated to buffers for a particular subpool might be better used elsewhere in CICS.

The CICS statistics provide information about the number of times an I/O request was made to a LSR pool, and identify the subpool to which the request was made.

CPExpert calculates the total I/O requests made to each LSR subpool that was defined. The resulting value is compared with the LSRUSEn guidance variable (where "n" is the number of the LSR pool, ranging from 1 to 8). CPExpert produces Rule CIC176 if the total number of I/O requests to the LSR subpool is lower than the LSRUSEn guidance variable.

The default value for each LSRUSEn guidance variable is based on the global LSRUSE guidance variable. The default values for the LSRUSE guidance variable is 100, meaning that CPExpert will produce Rule CIC176 when less than 100 I/O requests are made to the LSR subpool. You can use the default value for the LSRUSE guidance variable, specify a different value for this global variable (and have the global value applied to each LSR pool), or specify a different value for one or more LSR pools.

CPExpert will not produce Rule CIC176 unless the condition exists for more than 50% of the CICS statistics intervals being analyzed.

Suggestion: As mentioned above, if VSAM files do not use the LSR subpools that are created, the storage dedicated to a particular subpool might be better used elsewhere in CICS.

- If storage is a constraint to CICS performance, and Rule CIC176 is produced for a particular LSR subpool, you should consider (1) eliminating the LSR subpool so the storage can be used elsewhere in CICS, or (2) reducing the number of buffers assigned to the subpool. If you eliminate the LSR subpool, you should identify any VSAM files that have specified the LSR subpool and change their specification to a different LSR pool.

- You can alter CPExpert's analysis by specifying a different value for either the global LSRUSE guidance variable or use the LSRUSEn guidance variable to alter CPExpert's analysis for specific LSR pools.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 65-68, pages 232-238, and page 244.

CICS/MVS Version 2.1.2 Performance Guide: pages 158-162, page 170, and pages 394-397.

CICS/ESA Version 3.1.1 Performance Guide: pages 71-73, pages 93-106, and page 239.

CICS/ESA Version 3.2.1 Performance Guide: pages 147-152, page 155, and pages 310-321.

CICS/ESA Version 3.3.1 Performance Guide: pages 157-162, pages 165-166, and pages 329-339.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix A.1.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.9.

CICS/TS Release 1.2 Performance Guide: Section 4.4.2, Section 4.4.4, and Appendix 1.1.10.

CICS/TS Release 1.3 Performance Guide: Section 4.6.2, Section 4.6.4, and Appendix 1.1.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (VSAM resource usage (LSRPOOL)), Chapter 18 (VSAM buffer allocations for LSR), and Appendix A (Table 53).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.5.2 Defining VSAM resource usage, Section 4.5.4 Defining VSAM buffer allocations for LSR, and Appendix 1.1.17.6.

Thanks: Thanks to Paul Gordon (Bank of America) for suggesting this rule.

Rule CIC177: Large percent UPDATE option used without WRITE or DELETE

Finding: The CICS interval statistics showed that a large percent of the file control commands with an UPDATE option were issued against a VSAM data set, without a corresponding WRITE or DELETE file control command for the VSAM data set.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: A CICS application accesses VSAM data sets using CICS file control commands. These file control commands read records, browse records, write records, and delete records[1].

Optionally, the read commands can specify that an update is to occur with the record being read (keyword UPDATE). The UPDATE option guarantees read integrity for the record.

- One result of this UPDATE option is that the record is locked (and, depending on the type of file, the control interval is locked). CICS serializes updates to recoverable resources so that if a transaction fails, its changes to those resources can be backed out independently of those made by any other transaction. Consequently, a transaction updating a recoverable resource gets control of that resource until the transaction terminates or until the transaction indicates that it wants to commit those changes with a SYNCPOINT command. Other transactions requiring the same resource must wait until the first transaction finishes with the resource.

- Another result of this UPDATE option is that, if the VSAM data set is assigned to a CICS-maintained data table[2], the VSAM source data set must be referenced by VSAM before the record is referenced in the data table reference. That is, any update or delete action on a record in the data table is automatically applied to the source data set before being applied to the data table.

---

[1] These simple read, write, and delete file control actions have many variations, such as READNEXT, READPREV, etc.

[2] A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets.

Both results from using the UPDATE option cause overhead and potentially degrade performance. Consequently, the UPDATE option should be used only if the record is actually updated or it is deleted.

It is possible, of course, that the application logic cannot determine whether the record is to be updated or deleted until the record is read. In this case, the UPDATE option must be specified to preserve file integrity.

Some applications exercise the READ UPDATE option without considering the potential overhead or performance degradation, and some applications do not need to use the UPDATE option. For these applications, programming changes would be warranted if a large number of records were read with the UPDATE option, but those records were neither updated nor deleted.

File control statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed a VSAM data set using the UPDATE option but did not subsequently UPDATE or delete the records. CPExpert uses the following algorithm to calculate the percent unnecessary UPDATE option:

$$Percent\ unnecessary\ update\ option = \frac{Records\ updated + Records\ deleted}{GET\ UPDATE\ requests + BROWSE\ UPDATE\ requests}$$

where

Records updated	=	A17DSWRU
Records deleted	=	A17DSDEL+A17RMDEL
GET UPDATE requests	=	A17DSGU
BROWSE UPDATE requests	=	A17DSBRU (applies to VSAM RLS only)

Please refer to the CICS Performance Guides for a description of the individual variables.

CPExpert produces Rule CIC177 when the percent unnecessary UPDATE option is more than the value specified by the **PCTFCUPD** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTFCUPD** is 25 indicating that CPExpert should produce Rule CIC177 whenever more than 25% of the VSAM file accesses with the UPDATE option did not result in a corresponding change to the VSAM source data set.

CPExpert normally suppresses this finding if less than 500 GET UPDATE file control commands were issued against the file. You can specify a different threshold for suppressing Rule CIC177 by altering the **FCGETUPD** guidance variable in USOURCE(CICGUIDE).

Suggestion: If Rule CIC177 is consistently produced for a VSAM file, you consider the following alternatives:

- Consult with applications personnel to review the applications referencing the VSAM file. You should determine whether there is a logic or coding error that causes records to be read with the update option, but without a subsequent write or deletion of the record.

  As mentioned earlier, it is possible, that the application logic cannot determine whether the record is to be updated or deleted until the record is read. In this case, there is no error implied by the CIC177 finding, but the situation is simply a result of correct program logic.

- You can change the PCTFCUPD guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC177 is produced too often, or if application personnel are unable to take action.

- You can specify a different threshold for suppressing Rule CIC177 by altering the **FCGETUPD** guidance variable in USOURCE(CICGUIDE).

Reference: CICS/ESA Version 4.1.1
CICS Application Programming Guide: Section 3.1.6.
CICS Performance Guide: Appendix A.1.11.

CICS/TS Release 1.1
CICS Application Programming Guide: Section 3.1.6.
CICS Performance Guide: Appendix 1.1.9.

CICS/TS Release 1.2
CICS Application Programming Guide: Section 3.1.5.
CICS Performance Guide: Appendix 1.1.10.

CICS/TS Release 1.3
CICS Application Programming Guide: Section 4.1.7.
CICS Performance Guide: Appendix 1.1.11.

CICS/TS for z/OS Release 2.1
CICS Application Programming Guide: Section 4.1.7.
CICS Performance Guide: Appendix 1.1.11.

CICS/TS for z/OS Release 2.2
CICS Application Programming Guide: Section 4.1.7.
CICS Performance Guide: Appendix 1.1.11.

Rule CIC184: The size of the Temporary Storage data set may be too small

Finding: CPExpert has determined that the size of the Temporary Storage data set may be too small.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region, particularly with respect to individual tasks.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: If auxiliary temporary storage is not available when a task requests allocation, the task is either suspended because of a NOSPACE condition, or may have been forced to ABEND.

It is unlikely that a significant number of task ABENDs occur because of inadequate temporary storage. The program developers or production monitors probably would take immediate steps to prevent this situation. However, a task suspend because of inadequate temporary storage may not be so visible.

The number of times auxiliary temporary storage is exhausted is reported in the CICS statistics. CPExpert produces Rule CIC184 if this value is greater than zero when analyzing the daily statistics.

Suggestion: CPExpert suggests that you consider the following alternatives:

- If this is a production CICS region, immediately take steps to resolve the problem.

- If this is a test CICS region, determine whether the problem can be resolved by better testing procedures.

- Increase the amount of auxiliary storage allocated for temporary storage. The auxiliary storage allocated for temporary storage is based upon the data set referenced by the DFHTEMP DD name in the CICS Job Control Language.

- Examine the transactions experiencing suspends or ABENDs because of inadequate temporary storage for their requirements. These transactions may need to be modified to reduce their requirement. In any event, if the transactions are suspended, they retain their resources and thus may damage the overall performance of other transactions being processed by CICS. If the transactions ABEND, they have unnecessarily used resources up to the ABEND point. In either situation, overall CICS performance is harmed.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 79 and pages 264-268.

CICS/MVS Version 2.1.2 Performance Guide: pages 193-197 and page 408.

CICS/ESA Version 3.1.1 Performance Guide: page 125 and pages 259-263.

CICS/ESA Version 3.2.1 Performance Guide: pages 231-235 and page 339.

CICS/ESA Version 3.3.1 Performance Guide: pages 249-253 and page 357.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.1 and Appendix A.1.26.

CICS/TS Release 1.1 Performance Guide: Section 4.10.1 and Appendix 1.1.25.

CICS/TS Release 1.2 Performance Guide: Section 4.10.1 and Appendix 1.1.26.

CICS/TS Release 1.3 Performance Guide: Section 4.14 and Appendix 1.1.29.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS temporary storage) and Appendix A (Table 124).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.13.1 Tuning the use of CICS temporary storage and Appendix 1.1.29.

Rule CIC185: The number of Temporary Storage buffers may be too high

Finding: CPExpert believes that the number of temporary storage buffers may be too high.

Impact: This finding should normally have a fairly LOW IMPACT on the performance of the CICS region. However, the finding is a result of CPExpert detecting a CICS storage constraint. Consequently, the finding may have more indirect impact by freeing storage.

Logic flow: The following rule caused this rule to be invoked:

Rule CIC110: CICS encountered a Short-on-Storage condition

Discussion: Multiple VSAM buffers allow multiple VSAM control intervals to be available in storage at the same time. The CICS temporary storage programs can then respond to several requests for temporary storage, by using different buffers.

Using multiple VSAM buffers also increases the probability that the control interval required by any particular request will already be available in a buffer. From this view, buffers can be viewed as a "in storage" caching device, and can result in substantially fewer physical I/O operations required of VSAM.

If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and tasks are queued serially by temporary storage queue name. However, VSAM can continue servicing requests for other temporary storage queue names by other tasks.

It is generally desireable to minimize the number of times tasks must wait for access to a VSAM buffer. When the tasks wait, they continue to hold system resources (e.g., real storage), and may cause additional performance problems because the resources are not available to service other tasks.

However, a large number of buffers allocated to VSAM causes increased storage requirements. If the buffers are unused, the storage required to support the buffers is wasted.

The CICS statistics analyzed by CPExpert indicated that none of the temporary storage I/O operations waited for VSAM buffers. This usually indicates that more buffers are allocated than are necessary to service the requests for temporary storage. If more buffers are allocated than necessary, storage is wasted. Under many circumstances, this would not be a concern. However, the CICS region experienced a short-on-storage condition, so it is important to conserve the use of storage. Consequently, unnecessary buffers should not be defined.

Suggestion: CPExpert suggests that you consider decreasing the number of VSAM buffers allocated for temporary storage. The number of VSAM buffers that CICS allocates for temporary storage is specified by the second parameter of the TS operand in the System Initialization Table (SIT).

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 79 and pages 264-268.

CICS/MVS Version 2.1.2 Performance Guide: pages 193-197 and page 408.

CICS/ESA Version 3.1.1 Performance Guide: page 125 and pages 259-263.

CICS/ESA Version 3.2.1 Performance Guide: pages 231-235 and page 339.

CICS/ESA Version 3.3.1 Performance Guide: pages 249-253 and page 357.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.1 and Appendix A.1.26.

CICS/TS Release 1.1 Performance Guide: Section 4.10 and Appendix 1.1.25.

CICS/TS Release 1.2 Performance Guide: Section 4.10 and Appendix 1.1.26.

CICS/TS Release 1.3 Performance Guide: Section 4.14 and Appendix 1.1.29.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS temporary storage) and Appendix A (Table 124).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.13.1 Tuning the use of CICS temporary storage and Appendix 1.1.29.

Rule CIC186: The number of Temporary Storage buffers may be too low

Finding: CPExpert believes that the number of temporary storage buffers may be too low.

Impact: This finding should normally have a fairly LOW IMPACT on the performance of the CICS region. However, if the VSAM buffer wait occurs often, then this finding will have a MEDIUM IMPACT or HIGH IMPACT on performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: VSAM multiple buffers allow multiple VSAM control intervals to be available in storage at the same time. When multiple buffers are available for temporary storage, the CICS temporary storage programs can use different buffers to respond to several concurrent requests for temporary storage.

Using multiple VSAM buffers also increases the probability that the control interval required by any particular request will already be available in a buffer. From this view, buffers can be viewed as a "in storage" caching device, and can result in substantially fewer physical I/O operations required of VSAM.

If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and tasks are queued serially by temporary storage queue name. However, VSAM can continue servicing requests for other temporary storage queue names by other tasks.

It is generally desireable to minimize the number of times tasks must wait for access to a VSAM buffer. When the tasks wait, they continue to hold system resources (e.g., real storage), and may cause additional performance problems because the resources are not available to service other tasks.

The CICS statistics analyzed by CPExpert indicated that the percent of temporary storage I/O operations waiting for VSAM buffers was higher than the TSIOWAIT guidance variable in USOURCE(CICGUIDE). The default value for the TSIOWAIT guidance variable causes this rule to be produced when **any** temporary storage I/O operations waited for buffers. This usually means that additional buffers should be allocated for temporary storage.

Note that IBM's CICS Performance Guides suggest that up to 5% of temporary storage I/O operations waiting for buffers may be tolerable. However, IBMLINK Document Q467834 suggests that the number of temporary storage buffers should be increased if **any** buffer waits occur. CPExpert agrees with the IBMLINK Document.

Suggestion: CPExpert suggests that you consider increasing the number of VSAM buffers allocated for temporary storage. The number of VSAM buffers that CICS allocates for temporary storage is specified by the second parameter of the TS operand in the System Initialization Table (SIT).

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 79 and pages 264-268.

CICS/MVS Version 2.1.2 Performance Guide: pages 193-197 and page 408.

CICS/ESA Version 3.1.1 Performance Guide: page 125 and pages 259-263.

CICS/ESA Version 3.2.1 Performance Guide: pages 231-235 and page 339.

CICS/ESA Version 3.3.1 Performance Guide: pages 249-253 and page 357.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.1 and Appendix A.1.26.

CICS/TS Release 1.1 Performance Guide: Section 4.10 and Appendix 1.1.25.

CICS/TS Release 1.2 Performance Guide: Section 4.10 and Appendix 1.1.26.

CICS/TS Release 1.3 Performance Guide: Section 4.14 and Appendix 1.1.29.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS temporary storage) and Appendix A (Table 124).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.13.1 Tuning the use of CICS temporary storage and Appendix 1.1.29.

IBMLINK Document Q467834
TITLE: Questions about CICS Shutdown Statistics

Rule CIC187: The number of Temporary Storage strings may be too high

Finding: CPExpert believes that the number of temporary storage strings may be too high.

Impact: This finding should normally have a fairly LOW IMPACT on the performance of the CICS region. However, the finding is a result of CPExpert detecting a CICS storage constraint. Consequently, the finding may have more indirect impact by freeing storage.

Logic flow: The following rule caused this rule to be invoked:

Rule CIC110: CICS encountered a Short-on-Storage condition

Discussion: Temporary storage programs issue VSAM requests to perform I/O between the temporary storage buffers and the VSAM temporary storage data sets (defined by the DFHTEMP DD name). If multiple strings are available, multiple VSAM requests can be executed concurrently. If multiple VSAM requests are executed concurrently, the buffers will be transferred to DASD quicker, since several I/O requests can be outstanding at one time.

If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and is analyzed in Rules CIC185 and CIC186. However, the CICS statistics provide additional information about whether the VSAM strings might contribute to the unavailability of buffers. The statistics report the number of times a wait for buffer occurred and not all buffers were being serviced because of a lack of VSAM strings. This situation is called "wait on VSAM string".

If the number of strings is as high as the number of buffers, no CICS task will wait for a string (although the task may wait for a buffer if all buffers are full). This is because all buffers are being serviced by a VSAM string. However, if the number of VSAM strings is less than the number of buffers, not all buffers can be serviced if they are all full.

The CICS statistics analyzed by CPExpert indicated that none of the temporary storage I/O operations waited for VSAM strings. This indicates that there are more strings allocated than are necessary to service the buffers. If more strings are allocated than necessary, storage is wasted. Under many circumstances, this would not be a concern. However, the CICS region experienced a short-on-storage condition, so it is important to conserve the use of storage. Consequently, unnecessary strings should not be defined.

Suggestion: CPExpert suggests that you consider decreasing the number of VSAM strings allocated for temporary storage. The number of VSAM strings that CICS allocates for temporary storage is specified by the third parameter of the TS operand in the System Initialization Table (SIT).

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/MVS Version 2.1.2 Performance* Guide: pages 193-197 and page 408.

CICS/ESA Version 3.1.1 Performance Guide: page 125 and pages 259-263.

CICS/ESA Version 3.2.1 Performance Guide: pages 231-235 and page 339.

CICS/ESA Version 3.3.1 Performance Guide: pages 249-253 and page 357.

CICS/TS Release 1.1 Performance Guide: Section 4.10 and Appendix 1.1.25.

CICS/TS Release 1.2 Performance Guide: Section 4.10 and Appendix 1.1.26.

CICS/TS Release 1.3 Performance Guide: Section 4.14 and Appendix 1.1.29.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS temporary storage) and Appendix A (Table 124).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.13.1 Tuning the use of CICS temporary storage and Appendix 1.1.29.

Rule CIC188: The number of Temporary Storage strings may be too low

Finding: CPExpert believes that the number of temporary storage strings may be too low.

Impact: This finding should normally have a fairly LOW IMPACT on the performance of the CICS region. However, if the VSAM string wait occurs often, then this finding will have a MEDIUM IMPACT or HIGH IMPACT on performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: Temporary storage programs issue VSAM requests to perform I/O between the temporary storage buffers and the VSAM temporary storage data sets (defined by the DFHTEMP DD name). If multiple strings are available, multiple VSAM requests can be executed concurrently. If multiple VSAM requests are executed concurrently, the buffers will be transferred to DASD quicker, since several I/O requests can be outstanding at one time.

If no buffer is available when a task attempts to access temporary storage, the task must wait. This situation is called "wait on buffers" and is analyzed in Rules CIC185 and CIC186. However, the CICS statistics provide additional information about whether the VSAM strings might contribute to the unavailability of buffers. The statistics report the number of times a wait for buffer occurred and not all buffers were being serviced because of a lack of VSAM strings. This situation is called "wait on VSAM string".

If the number of strings is as high as the number of buffers, no CICS task will wait for a string (although the task may wait for a buffer if all buffers are full). This is because all buffers are being serviced by a VSAM string. However, if the number of VSAM strings is less than the number of buffers, not all buffers can be serviced if they are all full.

The CICS statistics analyzed by CPExpert indicated that the percent of temporary storage I/O operations waiting for VSAM strings was higher than the TSIOWAIT guidance variable in USOURCE(CICGUIDE). The default value for the TSIOWAIT guidance variable causes this rule to be produced when **any** of the temporary storage I/O operations waited for strings. This usually means that additional strings should be allocated for temporary storage.

Note that IBM's CICS Performance Guides suggest that up to 5% of temporary storage I/O operations waiting for strings may be tolerable.

However, IBMLINK Document Q467834 suggests that the number of temporary storage buffers should be increased if **any** buffer waits occur. CPExpert agrees with the IBMLINK Document, and applies the same reasoning to waits for temporary storage strings.

Suggestion: CPExpert suggests that you consider increasing the number of VSAM strings allocated for temporary storage. The number of VSAM strings that CICS allocates for temporary storage is specified by the third parameter of the TS operand in the System Initialization Table (SIT).

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 79 and pages 264-268.

CICS/MVS Version 2.1.2 Performance Guide: pages 193-197 and page 408.

\

CICS/ESA Version 3.1.1 Performance Guide: page 125 and pages 259-263.

CICS/ESA Version 3.2.1 Performance Guide: pages 231-235 and page 339.

CICS/ESA Version 3.3.1 Performance Guide: pages 249-253 and page 357.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.1 and Appendix A.1.26.

CICS/TS Release 1.1 Performance Guide: Section 4.10 and Appendix 1.1.25.

CICS/TS Release 1.2 Performance Guide: Section 4.10 and Appendix 1.1.26.

CICS/TS Release 1.3 Performance Guide: Section 4.14 and Appendix 1.1.29.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS temporary storage) and Appendix A (Table 124).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.13.1 Tuning the use of CICS temporary storage and Appendix 1.1.29.

IBMLINK Document Q467834
TITLE: Questions about CICS Shutdown Statistics

IBMLINK Document Q326699
TITLE: Implications of STRNO, STRNOG (Strings reserved for GET) in a VSAM-LSR context.

Rule CIC190: The number of DL/I Threads may be too low

Finding: CPExpert believes that the number of DL/I Threads may be too low.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: The DLTHRED operand in the System Initialization Table (SIT) specifies the number of concurrent DL/I threads that can be allocated for IMS/VS data bases. The number of concurrent DL/I threads limits the number of tasks concurrently scheduled for use of IMS/VS resources.

The DLTHRED operand is used at IMS/VS initialization to create control blocks. One set of control blocks is created for each DL/I thread. For each DL/I thread, control blocks are allocated for the CICS Interface Scheduling Block (ISB), the IMS/VS Parameter Block (PXPARMS), and the Partition Specification Table (PST). Storage for these control blocks is acquired at CICS initialization. The control blocks are retained while CICS is operational.

The amount of storage required for each DL/I thread depends upon the release of IMS. For each DL/I thread, IMS/VS 2.2 requires 9K bytes of storage and IMS/ESA requires 12K bytes of storage. A significant amount of real and virtual storage can be required if a large number of DL/I threads are specified. However, if the number of DL/I threads is too low, tasks can wait for threads. If tasks wait for DL/I threads, they tie up storage while they are waiting, and response suffers. Therefore, there is a tradeoff between the amount of storage allocated for DL/I threads and the delays caused by tasks waiting for threads.

CPExpert produces Rule CIC190 if storage was not a constraint and the number of times tasks waited for DL/I Threads was greater than the DLIWAIT guidance value. The default value of the DLIWAIT guidance parameter is zero.

Suggestion: CPExpert suggests that you consider increasing the number of DL/I Threads, as specified by the DLTHRED operand in the SIT. Tasks should not wait for DL/I threads unless storage is a constraint.

In any event, the DLTHRED value should be less than the AMXT value in the SIT[1]. Serious system degradation can occur if the DLTHRED value is higher than the AMXT value. (Refer to Rule CIC191 for an explanation of this situation.)

If storage **is** a constraint to the CICS region, you may wish to leave the DLTHRED value at its current setting. Depending upon the seriousness of the storage constraint, you may even wish to decrease the DLTHRED value so that some tasks wait on DL/I threads.

The CICS Performance Guides indicate that it may be acceptable for 5-10% of the tasks to wait on DL/I threads. For most environments, CPExpert believes that the 5-10% value is much too high. However, you may have a unique situation; if you **do** have a unique situation, you should change the CPExpert DLIWAIT guidance variable to prevent the spurious firing of this rule.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 69 and pages 255-256.

CICS/MVS Version 2.1.2 Performance Guide: pages 180-181 and pages 398-399.

CICS/ESA Version 3.1.1 Performance Guide: pages 55-60 and pages 249-250.

CICS/ESA Version 3.2.1 Performance Guide: pages 170-171 and pages 275-280.

[1]This comment does not apply with CICS/ESA Version 4.1, as the AMXT parameter is not used with this level of CICS.

CICS/ESA Version 3.3.1 Performance Guide: page 181 and pages 295-299.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.5.4 and Appendix A.1.5.

CICS/TS: not applicable.

CICS/TS for z/OS: not applicable.

Rule CIC191: The DLTHRED specification may be too high

Finding: CPExpert believes that the DLTHRED specification in the System Initialization Table (SIT) may be too high. Alternatively, the AMXT specification in the SIT is too low. In either case, there is a potentially serious performance problem caused by the two specifications.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region. This rule does not apply to CICS Version 4.1 and subsequent versions of CICS.

Logic flow: This is a basic finding, based upon an analysis of the CICS System Initialization Table.

Discussion: The DLTHRED operand in the SIT specifies the number of concurrent DL/I threads that can be allocated for IMS/VS data bases. The number of concurrent DL/I threads limits the number of tasks concurrently scheduled for use of IMS/VS resources.

The AMXT operand in SIT limits the total number of concurrent **active** tasks in the CICS region. All new and resumed tasks must pass the AMXT limit before CICS dispatches the tasks. Tasks are selected for dispatching only if fewer than AMXT tasks are already active. (The AMXT value actually specifies how far down the active chain CICS will scan, searching for a task to dispatch. Please refer to Rule CIC102 for additional information about the functioning of AMXT.)

If an active task waits for an IMS I/O operation, it may be inactive. If the number of DL/I threads is greater than the number of active tasks, the waiting task may not be dispatched when the IMS I/O operation completes because the number of active tasks may be at the AMXT threshold. This situation can arise even though only lower priority tasks are active. The response of the waiting task could be considerably lengthened in this event, particularly if VSAM control area splits can occur in the IMS/VS data base.

CPExpert produces Rule CIC191 if the value of the DLTHRED operand is not less than the value of the AMXT operand.

Suggestion: CPExpert suggests that you either decrease the DLTHRED value or increase the AMXT value. The DLTHRED value be less than the AMXT value in the SIT. Serious system degradation can occur if the DLTHRED value is not less than the AMXT value.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 255-256.

CICS/MVS Version 2.1.2 Performance Guide: pages 180-181.

CICS/ESA Version 3.1.1 Performance Guide: pages 249-250.

CICS/ESA Version 3.2.1 Performance Guide: pages 170-171.

CICS/ESA Version 3.3.1 Performance Guide: page 181.

CICS/ESA Version 4.1.1: not applicable.

CICS/TS: not applicable.

CICS/TS for z/OS: not applicable.

Rule CIC192: The DMBPL specification may be too low

Finding: CPExpert believes that the DMBPL specification in the System Initialization Table (SIT) may be too low.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: The DMBPL operand in the SIT specifies the number of blocks in data management block (DMB) pool. CICS uses the value specified in the DMBPL operand to limit the total virtual storage allocated at any one time to the DMB pool. CICS does not reserve the amount of storage specified, but allocates and deallocates the storage as required.

If tasks must wait for DMB pool space, data bases are closed to make room for the new DMB to be loaded, and the new data base will be opened. This process will halt processing in the entire CICS region until the CLOSE/OPEN processing completes. Severe response degradation would normally result from halting processing in the CICS region. Additionally, the CICS region could experience higher and highly erratic CPU time for CICS Task Control.

CPExpert produces Rule CIC192 if there were any waits for DMB pool space.

Suggestion: CPExpert suggests that you increase the value of the DMBPL operand in the SIT to eliminate the waits for DMB pool space.

- Since CICS does not reserve the virtual storage for the DMB pool space, there is little danger in specifying additional space. However, there may be significant performance degradation if insufficient space is specified.

- If virtual storage is a constraint, consider limiting the number of tasks using IMS/VS (assigning them to a Task Class and using the CMXT operand to limit the number of tasks). The CICS Performance Guide suggests that you particularly consider limiting lengthy tasks and heavy update tasks.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 69 and pages 257-258.

CICS/MVS Version 2.1.2 Performance Guide: pages 182-183 and pages 398-399.

CICS/ESA Version 3.1.1 Performance Guide: pages 55-60 and pages 250-252.

CICS/ESA Version 3.2.1 Performance Guide: pages 172-174 and pages 276-280.

CICS/ESA Version 3.3.1 Performance Guide: pages 182-184 and pages 296-299.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.5.5 and Appendix A.1.5.

CICS/TS: not applicable.

CICS/TS for z/OS: not applicable.

Rule CIC193: The IMS ENQPL storage pool specification may be too low

Finding: CPExpert believes that the ENQPL specification in the System Initialization Table (SIT) may be too low.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: The ENQPL operand in the SIT specifies the number of blocks in the IMS enqueue control block pool. The enqueue control block pool is used only if program-isolation scheduling is being used. If the ENQPL value is too small, the IMS/VS DB task abends with a U0775 pseudoabend message, causing dynamic backout of the changes.

As mentioned above, the ENQPL operand is used only with program-isolation scheduling. With CICS and IMS in a data-sharing environment, the Internal Resource Lock Manager (IRLM)[1] replaces the program isolation function. Consequently, it is not necessary to specify the ENQPL operand for the CICS region if IRLM is present.

CPExpert produces Rule CIC193 if the percent of the ENQ control block pool used was greater than the ENQPOOL guidance variable in CPEXPERT.USOURCE(CICGUIDE). The default for the ENQPL guidance variable is 75, indicating that Rule CIC193 will be produced when more than 75% of the ENQ control block pool is used.

The purpose of Rule CIC193 is to give an "early warning" of a potential problem. Obviously, if the IMS/VS DB task abends, action would be taken to increase the ENQPL value. Rule CIC193 is present to give an advance warning that such abends may occur with the present setting of the ENQPL operand.

Suggestion: CPExpert suggests that you increase the value of the ENQPL operand in the SIT to minimize the possibility of running out of enqueue pool space.

[1] Also known as the IMS Resource Lock Manager, IMS/VS Resource Lock Manager, or Inter-Region Lock Manager depending on the IBM document you reference.

With IMS 2.2, IMS acquires the enqueue pool above the 16megabyte line (the CICS 2.1.2 Performance Guide erroneously states that it still is acquired from OSCOR).

IMS uses only what is required during operation, rather than what is specified in the SIT. Consequently, there is little danger of specifying a very large value for the ENQPL operand (maximum value is 999, representing 999K of storage).

Alternatively, you should verify that batch IMS programs are issuing regular checkpoints (CHKPT) If the batch programs are not issuing regular checkpoints, they should be modified to issue them.

Alternatively, if you feel that this rule is produced prematurely, increase the value of the ENQPL guidance variable.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 69 and pages 257-258.

CICS/MVS Version 2.1.2 Performance Guide: pages 182-183 and pages 398-399.

CICS/ESA Version 3.1.1 Performance Guide: pages 55-60 and pages 250-252.

CICS/ESA Version 3.2.1 Performance Guide: pages 172-174 and pages 276-280.

CICS/ESA Version 3.3.1 Performance Guide: pages 182-184 and pages 296-299.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.5.5 and Appendix A.1.5.

CICS/TS: not applicable.

CICS/TS for z/OS: not applicable.

IMS/VS V.2 Installation Notebook, G320-9541-01

IBMLINK Document Source: ELSSECVM02FS
TITLE: CICS-DL/I use of CSA

Rule CIC194 The PSBPL storage pool specification may be too low

Finding: CPExpert believes that the PSBPL specification in the System Initialization Table (SIT) may be too low.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: The PSBPL operand in the SIT specifies the number of blocks in the CICS-DL/I program specification block (PSB) pool. The PSB pool is used only if the CICS-DL/I interface is used. CICS uses the value specified in the PSBPL operand to limit the total virtual storage allocated at any one time to the PSB pool. CICS does not reserve the amount of storage specified, but allocates and deallocates the storage as required.

If there is insufficient space in the PSB pool to handle PSB pool requests, an IMS routine is called to free the least-used buffers in the pool. In this case, the oldest PSB is deleted and the new PSB is loaded. The deletion (and particularly the loading) of PSBs slows the processing of tasks. Other tasks in the CICS region can proceed (unless they are dependent upon the task waiting for PSB pool space). However, the response of the task waiting is degraded during the wait. Additionally, processor resources are required to delete the old PSB and load the new PSB. These effects can be eliminated if sufficient space is allocated to the PSB pool.

Allocating additional space to the PSB pool can require additional virtual storage (although CICS allocates the storage only if required). Allocating additional space to the PSB pool may be undesirable if storage is a constraint.

CPExpert produces Rule CIC194 if the number of waits for PSB pool space was greater than the PSBWAIT guidance variable, and if storage was not a constraint. The default for the PSBWAIT guidance variable is 0, indicating that Rule CIC194 will be produced if there were waits for PSB pool space and storage was not constrained.

Suggestion: CPExpert suggests that you increase the value of the PSBPL operand in the SIT to minimize the possibility of running out of space in the PSB pool.

IMS uses only what is required during operation, rather than acquiring what is specified in the SIT. Consequently, there is little danger of specifying a larger value for the PSBPL operand, unless storage is constrained.

If you feel that this rule is produced prematurely, increase the value of the PSBWAIT guidance variable in CPEXPERT.USOURCE(CICGUIDE).

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPExpert performs an analysis of historical information before taking action.

- If this finding is based upon an analysis of historical data covering a prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 69 and pages 257-258.

CICS/MVS Version 2.1.2 Performance Guide: pages 182-183 and pages 398-399.

CICS/ESA Version 3.1.1 Performance Guide: pages 55-60 and pages 250-252.

CICS/ESA Version 3.2.1 Performance Guide: pages 172-174 and pages 276-280.

CICS/ESA Version 3.3.1 Performance Guide: pages 182-184 and pages 296-299.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.5.5 and Appendix A.1.5.

CICS/TS: not applicable.

CICS/TS for z/OS: not applicable.

Finding: CPExpert believes that the number of buffers specified for Transient Data destinations may be too low.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: Transient data is used for a variety of purposes within CICS (e.g., servicing request from CICS, servicing requests from user tasks, passing requests to the operating system, initiating tasks based on queue trigger level specifications, etc.). Transient data destinations are defined by the DFHDCT macros. Transient data may be directed to extrapartition destinations or to intrapartition destinations.

- Extrapartition destinations are used to specify data that is outside the CICS region, but which are allocated to CICS. Extrapartition destinations are used for two purposes:

  - Sending data outside the CICS region. For example, data created by a transaction may be sent to an extrapartition destination to be processed by a batch job.

  - Retrieving data from outside the CICS region. For example, input to a transaction may be received from a terminal, using an extrapartition destination.

  Extrapartition requests are processed by QSAM, and requests are issued by CICS (rather than by CICS tasks). Consequently, the entire CICS region can wait for completion of an extrapartition request if something delays the request.

- Intrapartition destinations are used for data that is to be stored temporarily. An intrapartition destination may be a terminal, a file, or another system. All intrapartition destinations are retained in a single VSAM data set. Consequently, intrapartition destinations require allocation of VSAM buffers and strings. The performance of intrapartition destinations is affected by the number of buffers and the number of strings.

A VSAM buffer is used to hold each VSAM control interval (CI) required to respond to a request for an intrapartition destination. The number of buffers is explicitly specified using the TD operand in the System Initialization Table (SIT). The default is three buffers for the intrapartition destinations (recall that all intrapartition destinations reside in a single VSAM file).

An intrapartition request is queued if (1) the CI required is already in use or (2) no buffers are available. Intrapartition requests are queued by destination. If multiple buffers are specified (as is the default), requests for other intrapartition destinations can proceed even though a queue might exist for a different destination.

Providing multiple buffers has another, potentially more important, benefit. This benefit arises because CICS retains several VSAM CIs in storage. The use of multiple buffers increases the probability that the CI required by a particular request is already available in a buffer. If the CI is in a buffer, no physical I/O operation will be required by VSAM. This can lead to a significant reduction in the number of VSAM I/O operations, resulting in better response to tasks and resulting in decreased use of system I/O resources.

If no VSAM buffers are available to handle an intrapartition destination request, the request must wait for a buffer. The CICS statistics report the number of intrapartition requests that must wait for VSAM buffers.

CPExpert produces Rule CIC210 if the CICS statistics revealed that intrapartition requests must wait for VSAM buffers, and if CICS did not exhibit a short-on-storage condition. CPExpert provides information regarding the CI size and the number of buffers.

Suggestion: CPExpert suggests that you consider increasing the number of buffers assigned to intrapartition transient data. This is accomplished by increasing the number of buffers specified in the TD operand of the SIT.

There is a tradeoff between improving transient data performance and the increased storage requirements implicit in increasing the number of VSAM buffers assigned to intrapartition transient data. If storage is not constrained, you should not allow intrapartition requests to wait for buffers. However, if storage is constrained, you may wish to accept some small number of waits for transient data buffers.

In general, it may be better to restrict the number of tasks processed by CICS (using MXT or CMXT) and minimize the time each task spends in the system, rather than allowing tasks to wait for buffers and use system resources while they wait.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 74-76, pages 269-273, and page 384.

CICS/MVS Version 2.1.2 Performance Guide: pages 198-202, page 345, and pages 403-405.

CICS/ESA Version 3.1.1 Performance Guide: pages 138-143 and pages 263-268.

CICS/ESA Version 3.2.1 Performance Guide: pages 235-241 and pages 349-353.

CICS/ESA Version 3.3.1 Performance Guide: pages 254-255 and pages 367-371.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.2 and Appendix A.1.32.

CICS/TS Release 1.1 Performance Guide: Section 4.10.3 and Appendix 1.1.29.

CICS/TS Release 1.2 Performance Guide: Section 4.10.3 and Appendix 1.1.30.

CICS/TS Release 1.3 Performance Guide: Section 4.14.3 and Appendix 1.1.33.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS Transient Data) and Appendix A (Table 139).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.13.3 (Optimizing the performance of the CICS transient data (TD) facility) and Appendix 1.1.33.

Rule CIC211 The number of Transient Data strings may be too low

Finding: CPExpert believes that the number of strings specified for Transient Data destinations may be too low.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: Transient data is used for a variety of purposes within CICS (e.g., servicing request from CICS, servicing requests from user tasks, passing requests to the operating system, initiating tasks based on queue trigger level specifications, etc.). Transient data destinations are defined by the DFHDCT macros. Transient data may be directed to extrapartition destinations or to intrapartition destinations.

- Extrapartition destinations are used to specify data that is outside the CICS region, but which are allocated to CICS. Extrapartition destinations are used for two purposes:

  - Sending data outside the CICS region. For example, data created by a transaction may be sent to an extrapartition destination to be processed a batch job.

  - Retrieving data from outside the CICS region. For example, input to a transaction may be received from a terminal, using an extrapartition destination.

- Intrapartition destinations are used for data that is to be stored temporarily. An intrapartition destination may be a terminal, a file, or another system. All intrapartition destinations are retained in a single VSAM data set. Consequently, intrapartition destinations require allocation of VSAM buffers and strings. The performance of intrapartition destinations is affected by the number of buffers and the number of strings.

  VSAM requires a string for each concurrent file operation to intrapartition destinations. The number of strings is explicitly specified using the TD operand in the System Initialization Table (SIT). The default is three strings for the intrapartition destinations (recall that all intrapartition destinations reside in a single VSAM file).

An intrapartition request is queued if the number of concurrent requests exceeds the number of available strings. The CICS statistics report the number of intrapartition requests that must wait for VSAM strings.

CPExpert produces Rule CIC211 if the CICS statistics revealed that intrapartition requests must wait for VSAM strings, and if CICS did not exhibit a short-on-storage condition. CPExpert provides information regarding the number of strings allocated for intrapartition destinations.

Suggestion: This performance constraint can be easily relieved by increasing the number of strings specified in the SIT. CPExpert suggests that you increase the number of strings assigned to intrapartition transient data.

Reference: *CICS/MVS Version 2.1.2 Performance* Guide: pages 198-202, page 345, and pages 403-405.

CICS/ESA Version 3.1.1 Performance Guide: pages 138-143 and pages 263-268.

CICS/ESA Version 3.2.1 Performance Guide: pages 235-241 and pages 349-353.

CICS/ESA Version 3.3.1 Performance Guide: pages 254-255 and pages 367-371.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.10.2 and Appendix A.1.32.

CICS/TS Release 1.1 Performance Guide: Section 4.10.3 and Appendix 1.1.29.

CICS/TS Release 1.2 Performance Guide: Section 4.10.3 and Appendix 1.1.30.

CICS/TS Release 1.3 Performance Guide: Section 4.14.3 and Appendix 1.1.33.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 26 (CICS Transient Data) and Appendix A (Table 139).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.13.3 (Optimizing the performance of the CICS transient data (TD) facility) and Appendix 1.1.33.

Finding: CPExpert believes that the buffer size allocated for the system log journal is too small.

Impact: This finding should normally have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The level of impact depend on the number of times the "buffer full" condition was experienced by the system log journal.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: CICS uses the system log journal to automatically log changes to recoverable resources. During emergency restart after an uncontrolled or immediate CICS termination, ICS uses the system log as the source of information for restoring resources. For systems prior to CICS Version 3.2.1, a single buffer is acquired for the system log journal (CICS Version 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

The BUFSIZE operand determines the maximum size of a physical journal block. A physical journal normally contains more than one logical journal record, however. Journal records are placed sequentially in the journal buffer until a physical journal block is written. A physical journal block is written under the following conditions:

- The SYSWAIT=STARTIO (the default specification) was entered in the DFHJCT macro and a synchronizing request was made to the journal from CICS management modules.

- The logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand in the DFHJCT macro (the default value of the BUFSUV operand is the BUFSIZE value). When the logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand, a physical block output is started.

Prior to CICS Version 3.2.1, the BUFSUV operand can be used to overlap physical output with placing of additional logical records into the buffer. For CICS Version 3.2.1, two buffers are acquired and overlap is managed by switching buffers.

Prior to CICS Version 3.2.1, a physical output is initiated when the buffer space used by logical records reached the BUFSUV value. CICS continues to add logical records to the buffer until the physical output is complete (or until the buffer space is used up). After the physical output is completed, the logical records added to the buffer are "shifted up" to the beginning of the buffer.

If the buffer is full, tasks must wait until the physical output completes. This wait time delays tasks and lengthens response. These response delays definitely should be prevented, particularly for the system log journal.

However, there is potentially a more serious effect for the overall CICS region when tasks wait for the system log journal. When tasks wait for the system log, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPExpert produces Rule CIC220 if there were any occasions in which the system log journal buffer was full, forcing an I/O operation.

Suggestion: CPExpert suggests that you increase the buffer size for the system log journal. This is accomplished by increasing the value of the BUFSIZE operand in the DFHJCT macro for the system log (JFILED=SYSTEM in the DFHJCT macro).

The buffer size should be increased until the "buffer full" condition is eliminated for the system log journal. Tasks wishing to write to the system log journal must wait if the system log journal buffer is full, with the potential performance degradation described above.

The value for the BUFSUV operand should not be specified for the system log journal (to allow it to default to the BUFSIZE value) or it should be explicitly specified as the same as the BUFSIZE value. Physical blocks are written from the system log journal based on synchronous output requests (or by filling the buffer, in which a "buffer full" condition is experienced). Consequently, the buffer should be large enough such that no "buffer full" condition occurs.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 81 and pages 274-276.

CICS/MVS Version 2.1.2 Performance Guide: pages 203-205 and page 452.

CICS/ESA Version 3.1.1 Performance Guide: pages 85-87 and pages 268-271.

CICS/ESA Version 3.2.1 Performance Guide: pages 182-184 and pages 304-305.

CICS/ESA Version 3.3.1 Performance Guide: pages 192-194 and pages 323-324.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.6.2 and Appendix A.1.17.

CICS/TS: not applicable

CICS/TS for z/OS: not applicable..

Finding: CPExpert believes that the buffer size allocated for a user journal may too small.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region or of individual tasks referencing the user journal. The level of impact depend on the number of times the "buffer full" condition was experienced by the user journal.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: A user journal may be used to record file control or message activity during CICS execution. CICS can be directed to perform automatic journaling by specifying entries in either the file control table or the program control, or users can provide explicit user journaling routines.

For systems prior to CICS Version 3.2.1, a single buffer is acquired for a user journal (CICS Version 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

The BUFSIZE operand determines the maximum size of a physical journal block. A physical journal normally contains more than one logical journal record, however. Journal records are placed sequentially in the journal buffer until a physical journal block is written. A physical journal block is written for a user journal under the following conditions:

- The SYSWAIT=ASIS operand was entered in the DFHJCT macro. CICS maintains a dynamic threshold called the "sliding buffer shiftup value" which is adjusted according to the rate of journal requests and the amount of data written. The intent is to achieve an output rate of approximately two blocks per second. IBM discourages use of the SYSWAIT=ASIS option for CICS versions prior to Version 3.2.1, and the option is not available for CICS Version 3.2.1.

- The logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand in the DFHJCT macro (the default value of the BUFSUV operand is the BUFSIZE value). When the logical records placed into the buffer use the amount of buffer space specified in the BUFSUV operand, a physical block output is started.

For CICS prior to Version 3.2.1, the BUFSUV operand can be used to overlap physical output with placing of additional logical records into the buffer. For CICS Version 3.2.1, two buffers are acquired and overlap is managed by switching buffers.

Prior to CICS Version 3.2.1, a physical output is initiated when the buffer space used by logical records reached the BUFSUV value. CICS continues to add logical records to the buffer until the physical output is complete (or until the buffer space is used up). After the physical output is completed, the logical records added to the buffer are "shifted up" to the beginning of the buffer.

If the buffer is full, tasks using the user journal must wait until the physical output completes. This wait time delays the tasks and lengthens response. These response delays should be prevented, unless there are system design reasons for regularly writing out the journal buffers.

Additionally, there is potentially a more serious effect for the overall CICS region when tasks wait for user journal, if the journal is used by several tasks. When tasks wait for a journal output, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPExpert produces Rule CIC221 if the number of times a buffer full" condition exceeded the CPExpert **JCBUFUL** guidance variable in CPEXPERT.USOURCE(CICGUIDE). This guidance variable is provided so that the rule will not be produced spuriously in those situations when system design decisions require user journals to be written frequently and the BUFSUV operand is insufficient to meet the requirement. The default value for the JCBUFUL guidance variable is zero, indicating that the buffer full condition should not be encountered for user journal logs. This default is set so low to ensure that you read this narrative and appreciate the potential performance implications of the buffer full conditions.

Suggestion: Unless you have unusual circumstances, CPExpert suggests that you increase the buffer size for the user journal. This is accomplished by increasing the value of the BUFSIZE operand in the DFHJCT macro for the user journal. If your system design considerations require the user journal be written frequently, change the CPExpert **JCBUFUL** guidance variable so that this rule is not produced spuriously.

If you do not wish to increase the buffer size, examine the BUFSUV value (pre-CICS Version 3.2.1).

- The default value for BUFSUV is the value specified for the BUFSIZE operand. This default means that the buffer will not be written until the buffer is full. However, if the BUFSUV value is less than the BUFSIZE operand value, the buffer will be written when the BUFSUV value is reached. Additional records will be placed into the buffer (as described above) while the physical block is written. Consequently, the output of the physical block will be overlapped with insertion of new logical records into the buffer.

- If the value for the BUFSUV operand is not specified or if the value for the BUFSUV operand is approximately the same as the BUFSIZE value, little or no overlap will occur between writing the physical block and insertion of new logical records. Tasks wishing to write to the user log journal must wait in this case, with the potential performance degradation described above.

- If the value for the BUFSUV operand is not specified for this user journal or if the value for the BUFSUV operand is approximately the same as the BUFSIZE value, you should specify a value for the BUFSUV operand. This value should be **less than** the BUFSIZE operand value. It is not possible to determine how much less the value should be on a general basis. However, the value should be sufficiently less that the "buffer full" condition does not arise.

  If necessary, you can experiment with successively lower values until the "buffer full" condition ceases. For example, you might set the BUFSUV value to be lower than the BUFSIZE value by an amount sufficient to accommodate 2 logical records. If that specification does not eliminate the "buffer full" condition, then you should continue lowering the BUFSUV value until the "buffer full" condition is eliminated.

  Note that lowering the BUFSUV value will cause the physical block size to be smaller, and more physical blocks will be written for a given number of journal records. This, of course, generates more I/O operations in writing the journal records and more I/O operations for the software processing the journal records. It probably is not wise to lower the BUFSUV value too much below the existing BUFSIZE value. Rather, you might wish to increase the BUFSIZE value and minimize the number of I/O operations.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 81 and pages 274-276.

CICS/MVS Version 2.1.2 Performance Guide: pages 203-205 and page 452.

CICS/ESA Version 3.1.1 Performance Guide: pages 85-87 and pages 268-271.

CICS/ESA Version 3.2.1 Performance Guide: pages 182-184 and pages 304-305.

CICS/ESA Version 3.3.1 Performance Guide: pages 192-194 and pages 323-324.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.6.2 and Appendix A.1.17.

CICS/TS: not applicable

CICS/TS for z/OS: not applicable..

Rule CIC222 The CICS journal buffer size may be too small (shift-ups)

Finding: CPExpert believes that the buffer size allocated for the system log journal may too small.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: A user journal may be used to record file control or message activity during CICS execution. CICS can be directed to perform automatic journaling by specifying entries in either the file control table or the program control, or users can provide explicit user journaling routines.

For systems prior to CICS Version 3.2.1, a single buffer is acquired for a user journal (CICS Version 3.2.1 acquires two buffers for each journal). The size of each journal buffer is specified in the BUFSIZE operand of the DFHJCT macro.

The BUFSIZE operand determines the maximum size of a physical journal block. A physical journal normally contains more than one logical journal record, however. Journal records are placed sequentially in the journal buffer until a physical journal block is written. A physical journal block is written for a user journal under the following conditions:

- The SYSWAIT=ASIS operand was entered in the DFHJCT macro. CICS maintains a dynamic threshold called the "sliding buffer shiftup value" which is adjusted according to the rate of journal requests and the amount of data written. The intent is to achieve an output rate of approximately two blocks per second.

  IBM discourages use of the SYSWAIT=ASIS option for CICS versions prior to Version 3.2.1, and the option is not available for CICS Version 3.2.1.

- The logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand in the DFHJCT macro (the default value of the BUFSUV operand is the BUFSIZE value). When the logical records placed into the buffer used the amount of buffer space specified in the BUFSUV operand, a physical block output is started.

 Revised: April, 1998

For CICS prior to Version 3.2.1, the BUFSUV operand can be used to overlap physical output with placing of additional logical records into the buffer. For CICS Version 3.2.1, two buffers are acquired and overlap is managed by switching buffers.

However, prior to Version 3.2.1, a physical output is initiated when the buffer space used by logical records reached the BUFSUV value. CICS continues to add logical records to the buffer until the physical output is complete (or until the buffer space is used up). After the physical output is completed, the logical records added to the buffer are "shifted up" to the beginning of the buffer.

If the buffer is full, tasks using the user journal must wait until the physical output completes. This wait time delays the tasks and lengthens response. These response delays should be prevented, unless there are system design reasons for regularly writing out the journal buffers.

Additionally, there is potentially a more serious effect for the overall CICS region when tasks wait for user journal, if the journal is used by several tasks. When tasks wait for a journal output, they will become dispatchable whenever the physical output completes and their records are placed in the buffer. CICS can enter a stress condition if a number of tasks are suddenly dispatchable and require resources.

CPExpert fires Rule CIC221 if the number of times a buffer full" condition exceeded the JCBUFUL guidance variable. This guidance variable is provided so that the rule will not fire spuriously in those situations when system design decisions require user journals to be written frequently and the BUFSUV operand is insufficient to meet the requirement.

Suggestion: Unless you have unusual circumstances, CPExpert suggests that you increase the buffer size for the user journal. This is accomplished by increasing the value of the BUFSIZE operand in the DFHJCT macro for the user journal. If your system design considerations require the user journal be written frequently, change the JCBUFUL guidance variable so that this rule does not fire spuriously.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 255.

CICS/ESA Version 3.2 Performance Guide: page 234.

CICS/ESA Version 3.3: not applicable.

CICS/ESA Version 4.1: not applicable.

CICS/TS: not applicable.

Rule CIC230 — The CICS Dynamic Transaction Backout buffer size may be too small

Finding: CPExpert believes that the buffer size allocated for the dynamic transaction backout buffer may too small.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: The dynamic log buffer stores backout information in the dynamic log for dynamic transaction backout purposes. The maximum size of the dynamic buffer is specified by the DBUFSZ operand in the System Initialization Table (SIT).

The initial size of the dynamic log buffer is one-half the amount specified by the DBUFSZ, for each transaction. CICS dynamically adjusts the size of the dynamic log buffer based upon the requirements of each transaction.

If the data recorded in the dynamic log exceeds the buffer size, records spill on to temporary storage. The temporary storage can come from main storage or from auxiliary storage for CICS Systems prior to CICS Version 2.1.1. After CICS Version 2.1.1, the temporary storage will spill only to expanded storage if expanded storage is available.

Some spilling of the dynamic log is an expected effect of the dynamic adjustment of the size of the dynamic log buffer. There is a trade-off between the amount of storage allocated for the dynamic log buffer and the overhead caused by spilling records to auxiliary (or expanded) storage. This trade-off varies from system to system. However, the IBM CICS Performance Guide suggests that the DBUFSZ operand should be increased if more than 20% of the records are spilled.

CPExpert produces Rule CIC230 if more than 20% of the records logged by the dynamic transaction backout process are spilled.

Suggestion: CPExpert suggests that you consider increasing the buffer size for the dynamic transaction backout log buffer. This is accomplished by increasing the value of the DBUFSZ operand in the SIT.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: page 83 and pages 330-331.

CICS/MVS Version 2.1.2 Performance Guide: page 412 and pages 269-270.

CICS/ESA Version 3.1.1 Performance Guide: page 66 and pages 317-319.

CICS/ESA Version 3.2.1 Performance Guide: pages 215-216 and page 286.

CICS/ESA Version 3.3.1 Performance Guide: pages 235-236 and page 304.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.7.14 and Appendix A.1.9.

CICS/TS: not applicable

CICS/TS for z/OS: not applicable..

Finding: CPExpert believes that the Region Exit Interval (ICV) specification may be inappropriate. This finding could result from the ICV specification itself, or from CPExpert detecting an apparent conflict between the ICV specification and the specification of another parameter.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region. However, the finding could periodically have a HIGH IMPACT on performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS will enter an operating system WAIT state whenever it cannot dispatch a task. The WAIT state may be caused by no work to be done (that is, no CICS tasks are in the system) or may be caused by all CICS tasks waiting on some event to occur (the tasks could be waiting for I/O operations to complete, for example).

Before CICS enters the operating system WAIT state, it issues an STIMER with an expiration of the value specified by the ICV parameter in the System Initialization Table (SIT). The STIMER ensures that CICS will regain control upon expiration of the ICV value, unless CICS regains control earlier.

Many events can cause CICS to exit from the WAIT state and resume processing before expiration of the ICV value. As examples, expiration of the terminal scan delay (ICVTSD) interval will cause CICS to attempt to process terminal output requests, completion of I/O operation will cause CICS to resume the task waiting on the I/O, etc.

For many CICS regions, these other events will be the controlling factor and the ICV value will have no effect whatsoever.

There are some events which do not result in CICS checking whether other work can be done. For example, MRO requests may be "batched" from a connected region (the MROBTCH value in the connected region specifies how many requests will be batched).

Batching MRO requests reduces CICS processor overhead since the processor time required for waiting and posting TCBs is reduced. Additionally, batching MRO requests may reduce differences in CICS response time between peak periods and non-peak periods. This effect occurs because some "batching" normally occurs during peak periods because of queueing for resources. Batching MRO requests would have little effect during peak periods, but would delay tasks during non-peak periods. If the ICV value is selected properly, the delay in non-peak periods could cause response time to be approximately the same as the delay during peak periods.

CICS normally will process the batched MRO requests only when the number of batched requests is as large as the MROBTCH value. However, CICS will process the batched MRO requests whenever the ICV value lapses. Consequently, it is important if MRO batching is used to specify a value for ICV which will ensure reasonable response time.

Suggestion: The suggestions offered by CPExpert depends upon whether the CICS region is a Test or Production region, and depends upon the MROBTCH specification and the ICV specification.

- **For a production CICS region: CPExpert detected MROBTCH set to default of 1 and ICV set to default of 1000**. Under these conditions, MRO requests are not being batched and the ICV default is too low.

  The main purpose of the ICV value in this situation is to retain CICS pages in central storage. CICS will reference many of the page frames upon expiration of the ICV interval. Consequently, these page frames will have a low Unreferenced Interval Count (UIC) and will not likely be stolen.

  However, normal CICS operation in a production environment will cause this effect (that is, CICS tasks will be executed frequently) or the ICVTSD value will cause the effect. The ICV value will have an effect only during period of inactivity.

  The overhead associated with the low ICV is unnecessary, since it is unlikely that the System High UIC will decrease to the point that page frames will be stolen from the CICS core code. With a ICV value of 1000, the System High UIC would have to be UIC=1 before CICS frames would be stolen. Such a low System High UIC would be cause for serious action by MVS and is unlikely to occur without immediate action by system programmers or system tuners. Consequently, CPExpert believes that the default of ICV=1000 is too low and this specification can result in unnecessary overhead during periods of low CICS activity.

  CPExpert suggests that ICV value be increased to 5000. Increasing the ICV value will save unnecessary overhead.

If expanded storage is online and expanded storage is not seriously constrained, the ICV value could be increased even higher than 5000 (for example, up to 20000). This higher specification is suggested to further reduce unnecessary overhead without incurring paging delays. Page frames stolen from the CICS region would be sent to expanded storage (unless expanded storage is constrained such that the Migration Age is low). Page faults resolved from expanded storage do not normally effect performance since the page fault resolution is so quick.

- **For a test CICS region: CPExpert detected MROBTCH set to default of 1 and ICV set to default of 1000**. Under these conditions, MRO requests are not being batched and the ICV default is too low. CPExpert suggests that the ICV value be increased to 20000.

  Test CICS regions are not generally heavily used and the ICVTSD value would not be a governing factor. It normally would be better to allow the central storage required by the CICS to be used for other workloads. Allowing the CICS test region to have a high ICV value would cause the Unreferenced Interval Count to rise when the CICS region was unused, page frames could be stolen from the CICS region if necessary, and the processor storage frames could be used by other workloads. Response is not normally critical for test CICS regions, so paging delays would not be a serious performance issue.

- **For either a production or test CICS region: CPExpert detected MROBTCH set larger than default of 1 and ICV set to default of 1000 or larger**. Under these conditions, MRO requests are being batched, and the ICV default may be too high.

  The main purpose of the ICV value in this situation is to ensure that response time is not unnecessarily delayed during periods of low activity, because of the batching of MRO requests.

  During periods of high activity, the MROBTCH value would be quickly reached and requests would not be delayed too long. In periods of low activity, the MROBTCH value might not be so quickly reached and requests could be delayed longer than management objectives. The ICV value can be used to prevent this situation.

  CPExpert suggests that the ICV value be reduced to 500. This will result in a maximum delay to batched MRO requests of 500 milliseconds and an average delay of much less, depending upon how many MRO requests were batched.

Warning: The value specified for the ICV parameter must not be greater than the value specified for the ICVS parameter. The ICVS parameter controls when CICS detects a system stall condition. After the ICVS time has lapsed, CICS will stall-purge the lowest priority task that is specified as stall-purgable. If the value specified for ICV is greater than the value specified for ICVS, CICS might not be able to detect a stall condition.

Reference: *CICS/MVS Version 2.1.2 Performance Guide*: pages 142-145, 215, and 297-299.

CICS/ESA Version 3.1.1 Performance Guide: pages 219-220, 278-280, and 329-331.

CICS/ESA Version 3.2.1 Performance Guide: pages 118-121, 136-139, and 222-223.

CICS/ESA Version 3.3.1 Performance Guide: pages 129-131, 146-149, and 240-241.

CICS/ESA Version 4.1.1 Performance Guide:Section 4.2.8, Section 4.3.8, and Section 4.8.4.

CICS/TS Release 1.1 Performance Guide: Section 4.2.8, Section 4.3.8, and Section 4.8.4.

CICS/TS Release 1.2 Performance Guide: Section 4.2.8, Section 4.3.8, and Section 4.8.5.

CICS/TS Release 1.3 Performance Guide: Section 4.2.8, Section 4.3.8, and Section 4.12.5.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 15 (MVS and DASD - Region exit interval), Chapter 18 (Terminal scan delay), and Chapter 24 (MRO and ISC - Batching requests).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.2.8 (Tuning the region exit interval), Section 4.3.8 (Adjusting the terminal scan delay), and Section 4.11.5 (Batching requests).

Additionally, please refer to the CPExpert MVS Component User Manual, Rules MVS400(series) and MVS500(series), for a discussion of central storage and expanded storage concepts. Alternatively, please refer to the papers at **www.cpexpert.com** where central storage and expanded storage are discussed.

Finding: CPExpert noticed that the default value for MROBTCH was specified in the System Initialization Table (SIT).

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. Additionally, benchmarks by IBM have shown about 12% decrease in processor overhead resulting from batching MRO requests.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS regions which use Multiple Region Operation (MRO) can queue MRO requests rather than sending the requests to the other region as the requests arrive. This queueing, or batching, of the MRO requests saves the overhead of posting and dispatching the region for each request.

Batching of MRO requests is accomplished using the MROBTCH parameter in the SIT. The MROBTCH parameter specifies the number of MRO requests which are to be accumulated before providing the set of requests to the receiving region. When the number of requests specified by the MROBTCH parameter has been accumulated, or when the time specified by the ICV parameter has lapsed, the region is started to process the requests.

With CICS Version 2, batching of MRO requests applied only to MRO requests. With CICS/ESA Version 3, batching of the requests was extended to include some non-MRO events, such as:

- VSAM physical I/O completion
- Subtasked request completion (if SUBTSKS=1 is specified)
- DL/I request completion implemented through DBCTL

Benchmarks executed by IBM have shown a significant (20%-30%) improvement in Internal Throughput Rate and about 12% decrease in processor utilization when using MROBTCH=6 and using MROLRM=YES, versus a base system with MROBTCH=1 and MROLRM=NO.

The base system values represent the defaults which IBM supplies with the SIT, for CICS Version 2 and Version 3. However, the IBM *CICS Performance Guides* recommend that the defaults be changed.

Suggestion: CPExpert suggests that you consider changing the MROBTCH value in the SIT from the default of MROBTCH=1 to MROBTCH=6.

CPExpert also suggests that you specify ISV=500 when you implement MRO batching, and that you specify MROLRM=YES. Please see Rule CIC250 and Rule CIC253 for discussion about this suggestion.

Reference: *CICS/MVS Version 2.1.2 Performance Guide*: pages 142-145, 215, 297-299, and 472-473. (Note that the referenced benchmark results are reported on pages 472-473.)

CICS/ESA Version 3.1.1 Performance Guide: pages 219-220, 278-280, and 329-331.

CICS/ESA Version 3.2.1 Performance Guide: pages 118-121, 136-139, and 222-223.

CICS/ESA Version 3.3.1 Performance Guide: pages 129-131, 146-149, and 240-241.

CICS/ESA Version 4.1.1 Performance Guide:Section 4.2.8, Section 4.3.8, and Section 4.8.4.

CICS/TS Release 1.1 Performance Guide: Section 4.2.8, Section 4.3.8, and Section 4.8.4.

CICS/TS Release 1.2 Performance Guide: Section 4.2.8 and Section 4.8.5.

CICS/TS Release 1.3 Performance Guide: Section 4.12.5.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 24 (MRO and ISC - Batching requests (MROBTCH)).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.11.5 (Batching requests).

Rule CIC252 The MROBTCH value may be too large

Finding: CPExpert believes that the MROBTCH value specified in the System Initialization Table (SIT) may too large.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS regions which use Multiple Region Option (MRO) can queue MRO requests rather than sending the requests to the other region as the requests arrive. This queueing, or batching, of the MRO requests saves the overhead of posting and dispatching the region for each request. Please see Rule CIC251 for additional discussion of this topic.

CPExpert notes that a value greater than 6 was specified for the MROBTCH parameter.

IBM's *CICS Performance Guides* recommend that MROBTCH not be set higher than 6, since the decreasing additional processor saving is unlikely to be worth the potentially increased response time.

Suggestion: CPExpert suggests that you consider changing the MROBTCH value in the SIT to MROBTCH=6.

Reference: *CICS/MVS Version 2.1.2 Performance Guide*: page 215.

CICS/ESA Version 3.1.1 Performance Guide: page 279.

CICS/ESA Version 3.2.1 Performance Guide: page 222.

CICS/ESA Version 3.3.1 Performance Guide: pages 240-241.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.8.4.

CICS/TS Release 1.1 Performance Guide: Section 4.8.4.

CICS/TS Release 1.2 Performance Guide: Section 4.8.5.

CICS/TS Release 1.3 Performance Guide: Section 4.12.5.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 24 (MRO and ISC - Batching requests (MROBTCH)).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.11.5 (Batching requests).

Rule CIC253 Default value was specified for the MROLRM keyword

Finding: CPExpert noticed that the default value for MROLRM was specified in the System Initialization Table (SIT).

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The life of mirror transactions in a MRO environment can be extended by specifying MROLRM=YES in the SIT. The default specification is MROLRM=NO.

- Setting MROLRM=NO causes the mirror transaction to be attached and detached for each function-shipped request until (1) the first request for a recoverable resource or (2) a file control Start Browse is received. After such a request is received, the mirror transaction remains attached to the session until the calling transaction reaches syncpoint.

- Setting MROLRM=YES in the CICS region which receives function shipping requests causes a mirror transaction to remain attached to the MRO session from first request until the calling transaction reaches syncpoint. IBM reports that this option has system-dependent effects:

  - Some systems show significant improvements in processor utilization per transaction when using MROLRM=YES. Systems which show the most improvement are typically systems with a significant percentage of inquiry transactions, where each transaction has multiple VSAM calls. Other systems which show significant improvement are transactions with many reads followed by a few updates.

  - Some systems show no performance improvement. This category may include workloads using IMS/ESA or transactions that significantly use VSAM update or browse.

  - Some systems could be degraded because there is an extra flow at syncpoint. For example, a system with a very simple inquiry transaction workload might show some degradation.

Beginning with the *CICS Performance Guide* for CICS Version 3.2, IBM recommends that the default be overridden to specify MROLRM=YES for regions receiving function shipping requests. While IBM began making this recommendation only in the CICS Version 3.2 *CICS Performance Guide*, the recommendation applies to CICS Versions prior to Version 3.2.

Suggestion: CPExpert suggests that you consider specifying MROLRM=YES in the SIT for this region.

Reference*:* *CICS/MVS Version 2.1.2 Performance Guide*: page 216.

CICS/ESA Version 3.1.1 Performance Guide: page 280.

CICS/ESA Version 3.2.1 Performance Guide: page 223.

CICS/ESA Version 3.3.1 Performance Guide: page 241.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.8.5.

CICS/TS Release 1.1 Performance Guide: Section 4.8.5.

CICS/TS Release 1.2 Performance Guide: Section 4.8.5.

CICS/TS Release 1.3 Performance Guide: Section 4.12.5.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 24 (MRO and ISC - Batching requests (MROBTCH)).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.11.5 (Batching requests).

Finding: CPExpert believes that the ISRDELAY value specified in the System Initialization Table (SIT) may be too low.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The ISRDELAY parameter in the SIT is the "intersystem refresh delay" specification. The parameter specifies how long entries can remain signed on to the remote system when you are running remote transactions over ISC and IRC links.

The default value for the ISRDELAY parameter is 30 minutes, indicating that a remote users will be signed off if there is no activity for 30 minutes. If a value of zero is specified, userid entries are deleted immediately after use. Once an entry is deleted, the user must sign on again to perform further activity. When the user signs on again, the external security manager (e.g., RACF) must be invoked.

Selecting an appropriate value for the ISRDELAY parameter is a tradeoff between:

- Management desire for security.
- The overhead required for the external security manager and the potential inconvenience or annoyance to the user.

Beginning with CICS/ESA Version 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the ISRDELAY parameter (A21SNTTM) and provide the average time that has elapsed between each reuse of userids (A21SNTAV). IBM suggests that the value for the ISRDELAY parameter should be increased if the number of "entries reused" (A21SNTRE) is low and the number of "entries timed out" (A21SNTTI) is high.

CPExpert first compares the average time that has elapsed between each reuse of userids (A21SNTAV) with the current value of the ISRDELAY parameter (A21SNTTM). CPExpert concludes that there is no continuing problem if the average reuse time is less than the ISRDELAY value.

If A21SNTAV is greater than A21SNTTM, then there may be a problem. The question is how serious is the problem (is it worth addressing), and does management really want the security processing to be performed after timeout. Since this is a function of installation management objectives, CPExpert provides a guidance variable in USOURCE(CICGUIDE) which is used to assess whether a problem exists. The guidance variable (**SNTCOUNT**) specifies the number of Signon Table (SNT) timeouts which are considered a problem.

CPExpert produces Rule CIC254 if A21SNTAV is greater than A21SNTTM, **and** the number of entries timed out (A21SNTTI) is greater than the SNTCOUNT guidance variable.

Suggestion: CPExpert suggests that you consider increasing the ISRDELAY to the value provided by Rule CIC254. The value provided by Rule CIC254 is simply the average time that has elapsed between each reuse of userids (A21SNTAV).

Alternatively, if you feel that Rule CIC254 is firing spuriously, please change the SNTCOUNT guidance variable in USOURCE(CICGUIDE).

Reference: CICS/ESA Version 3.2.1 Performance Guide, pages 302-303.

CICS/ESA Version 3.3.1 Performance Guide, page 61 and pages 321-322.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.24 and Appendix A.1.16.

CICS/TS Release 1.1 Performance Guide: Section 2.2.24 and Appendix 1.1.13.

CICS/TS Release 1.2 Performance Guide: Section 2.2.25 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 66).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Finding: CPExpert believes that the Persistent Verification Delay (PVDELAY) value specified in the System Initialization Table (SIT) may be too low.

Impact: This finding should normally have a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: With CICS/ESA Version 3.2.1, "persistent verification" is the term used to describe signing on to a remote system and having that sign on remain valid (or "persist") over multiple conversations until it is no longer needed.

By reducing the number of times a password is sent to the remote system, integrity is improved as the exposure of passwords is minimized. Performance is improved because CICS needs to call the external security manager (e.g., RACF) for password checking only during signon.

To sign onto CICS from a remote system, a valid userid and password must be provided in the attach to the remote system. The signon request is processed and, assuming that the userid and password are valid, CICS adds the userid to a table called the persistent verification "signed on from" list. The userid entry remains on the list until one of the following occurs:

- The user signs off.
- The entry is timed out (this timeout is controlled by the PVDELAY parameter).
- The connection between systems is lost.
- CICS is restarted.
- CICS receives an invalid attach from the userid.

As indicated above, the PVDELAY parameter in the SIT specifies how long userids are allowed to remain unused in the persistent verification "signed on from" list of a remote system.

The default value for the PVDELAY parameter is 30, indicating that an entry will be deleted if there is no activity for 30 minutes. If a value of zero is specified, userid entries are deleted immediately after use. Once an entry is deleted, the user must sign on again to perform further activity. When the remote user signs on again, the external security manager (e.g., RACF) must be invoked.

Selecting an appropriate value for the PVDELAY parameter is a tradeoff between:

- Management desire for security.

- The overhead required for the external security manager and the potential inconvenience or annoyance to the user.

Beginning with CICS/ESA Version 3.2.1, the CICS ISC/IRC Attach Time statistics provide the current value of the PVDELAY parameter (A21LUITM) and provide the average time that has elapsed between each reuse of entries in the "signed on from" list (A21LUIAV). IBM suggests that the value for the PVDELAY parameter should be increased if the number of "entries reused" (A21LUIRE) is low and the number of "entries timed out" (A21LUITI) is high.

CPExpert first compares the average time that has elapsed between each reuse of entries in the "signed on from" list (A21LUIAV) with the current value of the PVDELAY parameter (A21LUITM). CPExpert concludes that there is no continuing problem if the average reuse time is less than the PVDELAY value.

If A21LUIAV is greater than A21LUITM, then there may be a problem. The question is how serious is the problem (is it worth addressing), and does management really want the security processing to be performed after timeout. Since this is a function of installation management objectives, CPExpert provides a guidance variable in USOURCE(CICGUIDE) which is used to assess whether a problem exists. The guidance variable (**PVCOUNT**) specifies the number of persistent verification timeouts which are considered a problem.

CPExpert produces Rule CIC255 if A21LUIAV is greater than A21LUITM, **and** the number of entries timed out (A21LUITI) is greater than the PVCOUNT guidance variable.

Suggestion: CPExpert suggests that you consider increasing the PVDELAY to the value provided by Rule CIC255. The value provided by Rule CIC255 is simply the average time that has elapsed between each reuse of entries in the "signed on from" list (A21LUIAV).

Alternatively, if you feel that Rule CIC255 is firing spuriously, please change the PVCOUNT guidance variable in USOURCE(CICGUIDE).

Reference: *CICS/ESA Version 3.2.1 Performance Guide*: pages 302-303.

CICS/ESA Version 3.3.1 Performance Guide: page 61 and pages 321-322.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.24 and Appendix A.1.16.

CICS/TS Release 1.1 Performance Guide: Section 2.2.24 and Appendix 1.1.13.

CICS/TS Release 1.2 Performance Guide: Section 2.2.25 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.26 and Appendix 1.1.15

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC attach time entries) and Appendix A (Table 66).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Finding: CPExpert has detected that the CICS Monitoring Facility (Performance Class) was turned on in the System Initialization Table (SIT).

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. Turning on the CICS Monitoring Facility (Performance Class) data can result in a large number of records written to the SMF file and result in significant overhead. IBM estimates that the overhead can range from 5% to 10%, depending upon the workload.

Logic flow: This is a basic finding, based upon an analysis of the System Initialization Table.

Discussion: The CICS/ESA Monitoring Facility data are written based on the setting of the MN, MNPER, MNEXE, and MNEVE keywords in the CICS System Initialization Table. There are three types of CICS Monitoring Facility data: *performance* class data, *exception* class data, and *SYSEVENT* data.

- The performance class data consist of detailed information at the transaction level. At least one performance class record is written for each CICS transaction.

- The exception class data consist of information on exceptional conditions experienced by a CICS transaction. These exceptional conditions primarily include waits for storage or files.

- The SYSEVENT class data consist of information which is used primarily to record transaction timing data. The SYSEVENT class includes the terminal ID and elapsed time of each CICS transaction.

  The SYSEVENT class data are not applicable with CICS/ESA Version 4.1 if executing under the Workload Manager (MVS/ESA SP5 Goal Mode), as CICS provides the Workload Manager with detailed information about each transaction.

The CICS Monitoring Facility records can be extremely voluminous and can consume a significant part of the SMF file.

Additionally, collecting and recording the performance and accounting data can require a significant amount of processor resources. The IBM *CICS/ESA Performance Guides* estimate that the overhead is likely to be about 5% to 10%, depending upon the workload.

CPExpert produces Rule CIC256 when MN=ON and MNPER=ON have been specified in the SIT, and the CPExpert **PRODTEST** guidance variable in USOURCE(CICGUIDE) has been specified as "%LET PRODTEST=PROD" to indicate the that region is a production CICS region.

Suggestion: CPExpert suggests that you consider terminating the collection of CICS Monitor Facility (Performance Class) data. According to the IBM *CICS Performance Guides*, Performance Class data is rarely required for a production CICS region.

However, you should be aware that transaction accounting information is suppressed if CICS Monitor Facility (Performance Class) data is suppressed. Many sites use transaction accounting information for billing or charging purposes. You **should not** suppress CICS Monitor Facility (Performance Class) data if you use transaction accounting information.

Reference: *CICS/MVS Version 2.1.2 Performance* Guide: pages 49 and 208.

CICS/ESA Version 3.1.1 Performance Guide: pages 147 and 273.

CICS/ESA Version 3.2.1 Performance Guide: pages 53 and 241.

CICS/ESA Version 3.3.1 Performance Guide: pages 63 and 259.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.3 and Section 4.10.3.

CICS/TS Release 1.1 Performance Guide: Section 2.3 and Section 4.10.4.

CICS/TS Release 1.2 Performance Guide: Section 2.3 and Section 4.10.4.

CICS/TS Release 1.3 Performance Guide: Section 2.3 and Section 4.14.5.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 6 (The CICS monitoring facility).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.3 (The CICS monitoring facility)

Rule CIC257: CICS Auxiliary Trace was turned on

Finding: CPExpert has detected that the CICS Auxiliary Trace was turned on in the System Initialization Table (SIT) for a production CICS region.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region. Turning on the CICS Auxiliary Trace can result in significant overhead. IBM estimates that the overhead can be over 25%, depending upon the workload.

Logic flow: This is a basic finding, based upon an analysis of the System Initialization Table.

Discussion: The CICS Auxiliary Trace is a standard feature of CICS. The Auxiliary Trace gives an overview of transaction flows, and can be used to find situations that occur under full load.

The CICS Auxiliary Trace is intended to be used while debugging a CICS application or task. Auxiliary Trace can result in a **significant** overhead and it is not intended to be used in a normal production CICS region. Collecting and recording the performance and accounting data can require a significant amount of processor resources. The IBM *CICS Performance Guides* estimate that the overhead is likely to be over 25%, depending upon the workload.

CPExpert produces Rule CIC257 when AUXTR=ON has been specified in the SIT, and the CPExpert **PRODTEST** guidance variable in USOURCE(CICGUIDE) has been specified as "%LET PRODTEST=PROD" to indicate the that region is a production CICS region.

Suggestion: CPExpert suggests that you terminate CICS the Auxiliary Trace for a production CICS region.

Reference: *CICS/MVS Version 2.1.2 Performance* Guide: pages 67 and 211.

CICS/ESA Version 3.1.1 Performance Guide: pages 190 and 194.

CICS/ESA Version 3.2.1 Performance Guide: pages 26 and 98.

CICS/ESA Version 3.3.1 Performance Guide: pages 26 and 108.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.1 and Section 3.3.3.

CICS/TS Release 1.1 Performance Guide: Section 2.1 and Section 3.3.3.

CICS/TS Release 1.2 Performance Guide: Section 2.1 and Section 3.3.3.

CICS/TS Release 1.3 Performance Guide: Section 2.1 and Section 3.3.3.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 12 (CICS auxiliary trace).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.13.6 (CICS trace: performance considerations)

Rule CIC258: CICS Internal Trace Table might be too small

Finding: CPExpert has detected that the CICS Internal Trace Table might be too small.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. However, the finding can have a HIGH IMPACT on the site's ability to research problems and determine their causes.

Logic flow: This is a basic finding, based upon an analysis of the System Initialization Table.

Discussion: The CICS Auxiliary Trace is a standard feature of CICS. The Auxiliary Trace gives an overview of transaction flows, and can be used to find situations that occur under full load.

The CICS Auxiliary Trace is intended to be used while debugging a CICS application or task. Auxiliary Trace can result in a **significant** overhead and it is not intended to be used in a normal production CICS region. Collecting and recording the performance and accounting data can require a significant amount of processor resources. The IBM *CICS Performance Guides* estimate that the overhead is likely to be over 25%, depending upon the workload.

The CICS Auxiliary Trace uses the CICS Internal Trace Table to store trace entries. The default size of the Internal Trace Table is 16K (defined in the SIT). This size often is too small to allow enough trace entries to be stored so that a comprehensive amount of trace data is stored.

CPExpert produces Rule CIC258 when less than 100K has been specified for the TRTABSZ keyword in the SIT. While IBM recommends a much larger value (see below), CPExpert believes that your site might have deliberately selected a smaller value. Since you have changed the default value of 16K to a larger value, CPExpert will produce Rule CIC258 only when the specified value is not significantly higher than the default.

Suggestion: The IBM Support Center suggests that the default 16K size of the Internal Trace Table be increased significantly. IBM Support Center personnel indicate that "most shops" run with 1-2M trace tables.

Reference: *CICS/ESA Version 3.3.1 System Definition* Guide: Section 3.1.5.

CICS/ESA Version 4.1.1 System Definition Guide: Section 3.1.6.

CICS/TS Release 1.1 System Definition Guide: Section 3.1.5.

CICS/TS Release 1.2 System Definition Guide: Section 3.1.5.

CICS/TS Release 1.3 System Definition Guide: Section 3.1.5.

CICS/TS for z/OS Release 2.1 *System Definition* Guide: (The system initialization parameter descriptions - TRTABSZ keyword).

CICS/TS for z/OS Release 2.2 *System Definition* Guide: (The system initialization parameter descriptions - TRTABSZ keyword).

Thanks: David Ehresman (University of Louisville, KY) suggested this rule and provided the basic information. Thanks, David!

Rule CIC259: SUBTSKS=1 was specified in System Initialization Table

Finding: CPExpert noticed that subtasking was specified (SUBTSKS=1) in the System Initialization Table (SIT).

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The level of impact depends on whether the SUBTSKS=1 specification was correctly made, and whether sufficient processor resources exist to support the subtasking overhead.

Logic flow: This is a basic finding, based upon an analysis of the System Initialization Table.

Discussion: CICS always has two or three TCBs for normal processing, depending on the release of CICS:

- The quasi-reentrant (**QR** mode) TCB executes the quasi-reentrant application code and most CICS code. This TCB is available in all releases of CICS.
- The resource-owning (**RO** mode) TCB is used for opening and closing data sets and for program loading. This TCB is available in all releases of CICS.
- The file-owning (**FO** mode) TCB is used for opening and closing data sets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 1.

CICS optionally has other TCBs, again depending on the release of CICS:

- The concurrent **(CO** mode) TCB is responsible for VSAM subtasking. This TCB is available in all releases of CICS, but is active **only** if SUBTSKS=1 is specified.
- The Front End Programming **(SZ** mode) TCB is described in the CICS/ESA Front End Programming Interface User's Guide. This TCB is available in all releases of CICS, but is active **only** if FEPI=YES is specified in the SIT.
- The Open Network Computing Remote Procedure Call (**RP** mode) TCB is used to make ONC/RPC calls. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 3.

- A CICS task has a **J8** mode TCB if it needs to run a JAVA Virtual Machine. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 3.

- The **SO** mode TCB is used to make calls to the sockets interface of TCP/IP. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 3.

- The **SL** mode TCB is used to wait for activity on a set of listening sockets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 3.

- The **S8** mode TCB is used exclusively by a task if it needs to use the system Secure Sockets Layer. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 3.

- The **L8** mode TCB is identified in the *CICS Performance Guide* for OS/390, Release 3 (SC-33-16990-03 edition). However, the *Performance Guide* indicates that the J8 TCB is not in use for CICS/TS Release 3.

The objective of subtasks is to increase the maximum throughput of a single CICS system on multiprocessors. However, the intertask communication increases total processor utilization. Some of the subtasking is required to support specific CICS environments. However, the optional concurrent mode (CO) is implemented only if SUBTSKS=1 is specified in the SIT.

The default of the SUBTSKS parameter is SUBTSKS=0, which disallows concurrent mode TCBs. When SUBSTKS=1 is specified in the SIT, CICS takes the following actions:

- All Non -RLS VSAM file control WRITE requests to KSDS are subtasked.

- All other file control requests are never subtasked.

- Auxiliary temporary storage or intrapartition transient data requests are subtasked.

- Resource security checking requests are subtasked when the CICS main TCB (quasi-reentrant mode) exceeds approximately 70% processor activity.

IBM recommends that subtasking be used only in a multiprocessing system in a region that is limited by a single processor but has spare capacity on other processors in the MVS image. If subtasking is used in other circumstances, the subtasking can cause throughput degradation because of the dispatching of multiple tasks.

CPExpert produces Rule CIC259 when SUBTSKS=1 has been specified in the SIT. The primary reason for producing Rule CIC259 is to alert you to the possibility that significant throughput degradation might be experienced in the CICS region because of the specification.

Suggestion: CPExpert suggests that you consider the following questions:

- Is the CICS region operating in a multiprocessing environment (that is, will more than one engine be available for the CICS region)?
- Does the **main TCB** in the CICS region **and all equal or higher priority work** consume most of a single processor (at least 70% of the processor) during peak usage?
- Do other processors in the MVS image have spare CPU capacity that would be used by the subtask?
- Is there a significant amount of VSAM I/O activity occurring in support of the CICS region.

If the answers to **all** of the above question are "yes", then subtasking might be an appropriate approach.

However, if any answer to the above questions is "no" then subtasking probably is not a correct approach. Only if the above conditions are always true, the overhead associated with subtasking probably should not be used. In this case, CPExpert suggests that you change the SUBTSKS specification to disable subtasking of the concurrent TCB.

If you chose to continue with the SUBTSKS=1 specification in the SIT for this region, please "turn off" Rule CIC259 for the region (using the "turning off rules" process documented in Chapter 3.1 of Section 3 of the CICS Component User Manual.

Reference: *CICS/ESA Version 3.3.1 Performance Guide*: page 172.

CICS/ESA Version 4.1.1 Performance Guide: Section 4.4.11.

CICS/TS Release 1.1 Performance Guide: Section 4.4.11.

CICS/TS Release 1.2 Performance Guide: Section 4.4.11.

CICS/TS Release 1.3 Performance Guide: Section 4.5.11.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 18 (Subtasking VSAM).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Section 4.5.11 Permitting VSAM subtasking (SUBTSKS=1). |

Thanks: Bryant Osborn (Bank of America) suggested this rule and provided the basic information. Thanks, Bryant!

Rule CIC260: Insufficient sessions may have been defined

Finding: CPExpert believes that insufficient sessions may have been defined for the CICS region.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: A CICS region communicates with other CICS regions on the same system using the Multiple Region Operation (MRO) facility. The support within CICS that provides region-to-region communication is termed Interregion Communication (IRC). IRC is implemented by either (1) a CICS-supplied interregion program running in Supervisor state, or (2) cross-memory services provided by MVS.

A CICS region communicates with other systems using data formats and protocols embodied in IBM Systems Network Architecture (SNA). The support within CICS that provides region-to-system communication is termed Intersystem Communication (ISC). Within the following discussion, the CICS region which initiates communication is termed the "front-end" system and the other region/system is termed the "back-end" system.

A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system. The definition of a communication link to a remote system consists of two basic parts:

- **The definition of the remote system itself**. The remote system is defined by the DEFINE CONNECTION command, using the Resource Definition Online (RDO) facility. The remote system is defined by the DFHTCT TYPE=SYSTEM macro, if the macro-level resource definition facility is used.

- **The definition of sessions with the remote system**. The sessions with the remote system are defined by the DEFINE SESSIONS command, using the Resource Definition Online (RDO) facility. The sessions are defined by the DFHTCT TYPE=SYSTEM macro, if the macro-level resource definition facility is used.

Regardless of the method used to define sessions, sessions are defined as "send" sessions and "receive" sessions. The concept of send sessions and receive sessions differs depending upon whether IRC or ISC is used for the communication.

- **IRC communication**. With interregion communication, each session on the link is characterized as either a SEND or RECEIVE session. SEND sessions are used to carry an **initial** request from the local to the remote region and to carry any other data flows associated with the initial request. RECEIVE sessions are used to receive an **initial** request from the remote system.

  There can be a different number of SEND sessions and RECEIVE sessions. However, session types must be consistent between the local and remote regions. That is, there must be as many RECEIVE sessions defined on the remote region as there are SEND sessions defined on the local region, and there must be as many SEND sessions defined on the local system as there are RECEIVE sessions defined on the remote system.

  After the initial request has been made and received, the regions can communicate without regard to the SEND/RECEIVE characteristics of the session.

- **ISC communication (LU6.1)**. With intersystem LU6.1 communication, each session on the link is characterized as either a SEND or RECEIVE session. A SEND session is one in which the local CICS is the secondary and is the contention winner. SEND sessions can initiate communication at any time. A RECEIVE session is one in which the local CICS is the primary and is the contention loser. RECEIVE sessions can initiate communication only after "bidding" for permission to begin a bracket. To avoid the overhead of bidding, the number of SEND and RECEIVE sessions should be consistent with the expected flow between the two systems.

- **ISC communication (LU6.2)**. With intersystem LU6.2 communication, sessions are grouped into sets of sessions, referred to as modegroups. The sessions in each modegroup have identical characteristics, except the sessions are designated as contention winners or contention losers. The SESSIONS definition (for RDO definition) or DFHTCT TYPE=MODESET definition (for macro level definition) specifies the maximum number of contention winners in each modegroup. CICS determines which sessions are to be contention winners or contention losers when the sessions are bound.

A contention loser session can be converted to a contention winner session, if necessary, during operation. This conversion requires additional overhead (the bidding process must be implemented). The overhead should be avoided, if possible, by specifying a correct number of contention winners in the MAXIMUM sessions definition (for RDO definition) or in the MAXSESS definition (for macro level definition).

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

Note that a transaction could remain suspended indefinitely, waiting allocation for a session. The DTIMOUT parameter in DFHPCT TYPE=ENTRY definition can be used to limit the amount of time a transaction is suspended; CICS will terminate a transaction which has been suspended longer than the DTIMOUT value. The DTIMOUT specification can be used to prevent deadlock situations.

With LU6.2, the ALLOCATE command may request allocation from a specific session modegroup (with its own characteristics), or may not request any particular session modegroup. If a specific session modegroup is requested, CICS will restrict its allocation attempt to that modegroup. If no particular session modegroup is requested, CICS will attempt to allocate a session from any modegroup, **but selects modegroups in the order in which the modegroups are defined**.

This last point may be significant from a performance viewpoint. Rule CIC262 describes the implications of this allocation process in more detail.

With LU6.2, if CICS queues the ALLOCATE request, CICS attempts to make a session available by alternate means.

- CICS determines whether any unbound contention winner sessions exist. If an unbound contention winner session exists, the session is bound and allocated.

- CICS determines whether any unbound indeterminate sessions exist. Indeterminate sessions may exist when the total number of contention winner sessions defined on both the front-end and back-end is less than the total number of sessions defined. If an unbound indeterminate session is found, CICS declares it to be a contention winner session. The session is then bound and allocated.

- CICS determines whether any bound contention loser sessions exist but are unallocated. If a bound contention loser session exists, a bid is issued to the back-end, and the session is allocated after permission is obtained.

Some queuing for allocation requests may be unavoidable because an installation may have deliberately restricted the number of sessions to minimize resource use by CICS. The CICS/ESA Version 3.3.1 Performance Guide lists a number of effects potentially caused by increasing the number of sessions:

- The amount of real and virtual storage required to support CICS may increase.

- The use of storage on GATEWAY NCPs in the network may increase.

- The use of storage by VTAM may increase.

- The line loading in the network may increase.

- The back-end CICS system may not be able to cope with increased workload from the front-end system.

- The increased control block scanning by CICS may degrade performance.

In fact, some installations deliberately restrict the number of sessions as a means of controlling the number of CICS tasks (rather than using the TCLASS control method).

The decision of whether to increase the number of sessions should be viewed as a tradeoff between the potential disadvantages, versus the advantages of less task suspension, faster response, shorter transaction life, earlier release of resources, etc.

From a performance view, the ISC/IRC statistics provide information which can be used to assess whether enough sessions have been defined, whether the balance between contention winner sessions and contention loser sessions is appropriate, and whether there conflicting usage between APPC modegroups.

CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **generic** sessions in Rule CIC160. CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **specific** modegroups in Rule CIC261.

For Rule CIC260, CPExpert performs the following analysis:

- CPExpert evaluates the Peak Outstanding Allocates (A14ESTAM). A14ESTAM contains a count of the maximum number of ALLOCATE requests which were queued by CICS when a session could not be allocated.

  Note that if the NOQUEUE option were exercised by a user, unsatisfied ALLOCATE requests would not be included in this count. Unsatisfied ALLOCATE requests which were returned to the user with a SYSBUSY indicator would be included in Failed Allocates Due to Sessions in Use (A14ESTAO for generic requests). (Please refer to Rule CIC167.)

- CPExpert detects a **potential** problem when the maximum number of queued ALLOCATES for generic session allocation requests is greater than the **ALLOCQ** guidance variable in USOURCE(CICGUIDE). The default specification is **ALLOCQ=1**, indicating that Rule CIC260 would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available. **This low default value is intended only to alert you to a potential problem with the number of sessions defined, and is intended to make you aware of this analysis mechanism**.

  For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests queued. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few queued requests, but are concerned about the situations when hundreds or thousands of requests are queued.

Suggestion: CPExpert suggests that you consider increasing the number of sessions defined. Please refer to the above discussion to assess whether the number of sessions should be increased.

Alternatively, you may need to change the distribution of contention winners versus contention losers in LU6.1 or LU6.2 environments. Rule CIC163 provides more discussion of this alternative.

Alternatively, change the **ALLOCQ** guidance variable to cause CPExpert to signal a potential problem only when you view the problem as serious.

Reference: *CICS/MVS Version 2.1.2 Intercommunication Guide*.

CICS/ESA Version 3.1.1 Intercommunication Guide.

CICS/ESA Version 3.2.1 Intercommunication Guide.

CICS/ESA Version 3.3.1 Intercommunication Guide.

CICS/VS Multiple Region Planning Guide.

CICS MRO Tuning and Performance Guide.

CICS/ESA Version 3.1.1 Performance Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: pages 55-56 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 66).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC261: APPC modegroup may have insufficient sessions defined

Finding: CPExpert believes that there may be insufficient sessions defined for an APPC (LU6.2) modegroup.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

With LU6.2, the ALLOCATE command may request allocation from a specific session modegroup (with its own characteristics), or may not request any particular session modegroup. If a specific session modegroup is requested, CICS will restrict its allocation attempt to that modegroup. If no particular session modegroup is requested, CICS will attempt to allocate a session from any modegroup, but selects modegroups in the order in which the modegroups are defined. Rule CIC262 describes the implications of this allocation process in more detail.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

Some queuing for allocation requests may be unavoidable because an installation may have deliberately restricted the number of sessions to minimize resource use by CICS. The *CICS/ESA Version 3.3.1 Performance Guide* lists a number of effects potentially caused by increasing the number of sessions:

- The amount of real and virtual storage required to support CICS may increase.

- The use of storage on GATEWAY NCPs in the network may increase.

- The use of storage by VTAM may increase.

- The line loading in the network may increase.

- The back-end CICS system may not be able to cope with increased workload from the front-end system.

- The increased control block scanning by CICS may degrade performance.

In fact, some installations deliberately restrict the number of sessions as a means of controlling the number of CICS tasks (rather than using the TCLASS control method).

The decision of whether to increase the number of sessions should be viewed as a tradeoff between the potential disadvantages, versus the advantages of less task suspension, faster response, shorter transaction life, earlier release of resources, etc.

CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **generic** sessions in Rule CIC160. CPExpert detects the potential problem caused by queuing session ALLOCATE requests for **specific** modegroups in Rule CIC261.

For Rule CIC261, CPExpert performs the following analysis:

- CPExpert evaluates the Peak Outstanding Allocates for the modegroup (A20ESTAM). A20ESTAM contains a count of the maximum number of ALLOCATE requests which were queued by CICS when a session could not be allocated for the modegroup.

  Note that if the NOQUEUE option were exercised by a user, unsatisfied ALLOCATE requests for the modegroup would not be included in this count. Unsatisfied ALLOCATE requests which were returned to the user with a SYSBUSY indicator would be included in Failed Allocates Due to Sessions in Use (A20ESTAO for specific requests). (Please refer to Rule CIC168.)

- CPExpert detects a **potential** problem when the maximum number of queued ALLOCATES for specific modegroup session allocation requests is greater than the **ALLOCQ** guidance variable in USOURCE(CICGUIDE).

The default specification is **ALLOCQ=1**, indicating that Rule CIC261 would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available for a specific modegroup. **This low default value is intended only to alert you to a potential problem with the number of sessions defined, and is intended to make you aware of this analysis mechanism**.

For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests queued. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few queued requests, but are concerned about the situations when hundreds or thousands of requests are queued.

Suggestion: CPExpert suggests that you consider increasing the number of sessions allocated to the modegroup identified as having too few sessions. Please refer to the above discussion to assess whether the number of sessions should be increased.

Alternatively, you may need to change the distribution of contention winners versus contention losers for the modegroup.

Alternatively, you may need to change the order in which the modegroups are defined because of conflicting usage between modegroups. (Please refer to Rule CIC262.)

Alternatively, change the **ALLOCQ** guidance variable to cause CPExpert to signal a potential problem only when you view the problem as serious.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: pages 55-56 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC262: Conflicting usage between APPC modegroups may exist

Finding: CPExpert believes that there may be conflicting usage between APPC (LU6.2) modegroups.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

With LU6.2, the ALLOCATE command may request allocation from a specific session modegroup (with its own characteristics), or may not request any particular session modegroup. If a specific session modegroup is requested, CICS will restrict its allocation attempt to that modegroup. If no particular session modegroup is requested, CICS will attempt to allocate a session from any modegroup, **but selects modegroups in the order in which the modegroups are defined**. This last point may be significant from a performance viewpoint.

A situation can occur in which the modegroup most often requested in a **specific** session allocation request has been defined as the first modegroup in the modegroup definitions. Sessions from this modegroup will be allocated whenever a **generic** (non-specific) session allocation request is received by CICS. If a number of generic session allocation requests were received, all available sessions in the modegroup could be allocated. When a **specific** allocation request for a session from the modegroup was received, CICS would be unable to honor the allocation request and would queue the request (suspending the transaction).

This example refers to the first modegroup only for illustration purposes. CICS selects modegroups in the order in which they are defined, so the discussion would apply to the second or subsequent modegroups if all sessions in the first modegroup were allocated.

CPExpert detects this potential problem when Peak Outstanding Allocates (A20ESTAM) is non-zero for a particular modegroup. This situation indicates that some allocation requests were queued for the modegroup. This, by itself, could indicate that insufficient sessions were defined for the modegroup and would be analyzed by Rule CIC260.

However, CPExpert also noticed that the number of **generic** allocation requests (A14ESTAG) was much larger than the total number of **specific** allocation requests (A20ESTAS) for all modegroups during the interval covered by the CICS statistics. These values **may** indicate that the allocation delays for specific modegroups results from sessions in the modegroup being used to satisfy generic allocation requests. If this is the case, simply reordering the modegroups may solve the problem.

Suggestion: CPExpert suggests that you consider reordering the definition of APPC modegroups.

Modegroups for a connection are represented by TCT mode entries (TCTMEs). The modegroup name is taken from the MODENAME specified in the SESSIONS definition. The order of the TCTMEs is determined by the order in which CICS installs the SESSIONS definition. CICS installs the SESSIONS definitions based on the order of the SESSIONS names stored in the CICS System Definition (CSD) file. The SESSIONS names are stored in ascending alphanumeric sequence. Consequently, you must change the names of the SESSIONS definitions to change the order of the TCTMEs.

You can use the CEDA RENAME command with the AS option to rename the definition with a different SESSIONS name within the CSD group. By managing the order in which the TCTMEs are created, you can ensure that **specific** allocations reference modegroups lower down the TCTME chain. This will avoid conflicts with the **generic** allocates.

Alternatively, you can make all allocates specific to a particular modegroup.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: page 58 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC263: Balance between contention winners and contention losers may be inappropriate

Finding: CPExpert believes that the balance between contention winners and contention losers may need to be adjusted.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available.

With LU6.1 and LU6.2, CICS will attempt to make a session available if the ALLOCATE request cannot be honored. If there are no contention winner sessions available, CICS may issue a bid for an available contention loser session. If the bid is honored, the contention loser session is treated as a contention winner session and the session is allocated. Note that this does not apply to MRO connections.

Bidding for contention loser sessions requires unnecessary overhead and the process delays transactions. The overhead can be eliminated by specifying additional contention winner sessions. This can be done by specifying more sessions or by changing the balance between contention winners and contention losers.

CPExpert detects that there is an imbalance between LU6.1 contention winners andcontention losers by analyzing Peak Bids in Progress (A14EBHWM). Rule CIC263 is produced if CPExpert detects a problem with the number of generic sessions defined. (Rule CIC263 applies only to LU6.1 connections.)

Suggestion: CPExpert suggests that you consider making more contention winner sessions available. Making more contention winner sessions available should eliminate the need for CICS to bid for contention losers sessions to satisfy ALLOCATE requests.

More contention winner sessions can be made available by increasing the number of sessions or increasing the number of sessions defined as contention winners. Please refer to Rule CIC160 for a discussion of the implications of defining more sessions.

You must, of course, maintain consistency between the front-end and back-end definitions.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: page 57 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC264: Balance between contention winners and contention losers may be inappropriate for an APPC modegroup

Finding: CPExpert believes that the balance between contention winners and contention losers may need to be adjusted for an APPC (LU6.2) modegroup.

Impact: This finding should normally have a MEDIUM IMPACT on the performance of the CICS region. However, the finding could have a HIGH IMPACT on the performance of individual transactions if these transactions are queued for lengthy intervals.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions can acquire the use of a specific session in an APPC (LU6.2) environment by using the ALLOCATE command to request a specific modegroup. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in the specified modegroup in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available.

CICS will attempt to make a session available if the ALLOCATE request cannot be honored. If there are no contention winner sessions available in the modegroup, CICS may issue a bid for an available contention loser session in the modegroup. If the bid is honored, the contention loser session is treated as a contention winner session and the session is allocated.

Bidding for contention loser sessions requires unnecessary overhead and the process delays transactions. The overhead can be eliminated by specifying additional contention winner sessions. This can be done by specifying more sessions for the modegroup or by changing the balance between contention winners and contention losers in the modegroup.

CPExpert detects that there is an imbalance between APPC (LU6.2) contention winners and contention losers assigned to a modegroup by analyzing Peak Bids in Progress (A20EBHWM) for each modegroup. Rule CIC264 is produced if CPExpert detects a problem with the number of sessions defined for a modegroup.

Suggestion: CPExpert suggests that you consider making more contention winner sessions available in the modegroup identified by Rule CIC164. Making more contention winner sessions available should eliminate the need for CICS to bid for contention losers sessions to satisfy ALLOCATE requests for the modegroup.

More contention winner sessions can be made available by increasing the number of sessions on the MAXIMUM keyword of the DEFINE SESSIONS definition or by increasing the number of sessions defined as contention winners (the second parameter of the MAXIMUM keyword). Please refer to Rule CIC160 for a discussion of the implications of defining more sessions.

You must, of course, maintain consistency between the front-end and back-end definitions.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: page 57 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC265: Connection problems caused ALLOCATE requests to fail

Finding: CPExpert noticed that ALLOCATE requests failed because of problems with the communication link between the two systems.

Impact: This finding should normally have a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system.

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated.

CICS will process the ALLOCATE request only if the communication link is established and operational between the two systems. CICS will reject the ALLOCATE request if the connection has been released, the connection is out of service, or the mode group has been closed.

CPExpert produces Rule CIC265 if the number of Failed Link Allocates (A14ESTAF) is greater than the **ALLOCF** guidance variable. This includes requests to allocate **generic** sessions for APPC connections.

The default specification for the ALLOCF guidance variable is **%LET ALLOCF = 0**, indicating that no failed allocation requests are acceptable.

Suggestion: CPExpert suggests that you examine the state of the connection that CICS is trying to allocate a session on. Resolve any problem causing the ALLOCATE requests to fail.

You should check the CSMT log for the CICS statistics interval reported by Rule CIC265 for an indication of the problem with the connection.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: page 60 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Finding: CPExpert noticed that **specific** ALLOCATE requests failed because of problems with the communication link between the two systems.

Impact: This finding should normally have a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: A CICS region can communicate with another CICS region or with a remote system only after a "communication link" has been established between the CICS region and the other CICS region or the remote system.

Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated.

CICS will process the ALLOCATE request only if the communication link is established and operational between the two systems. CICS will reject the ALLOCATE request if the connection has been released, the connection is out of service, or the mode group has been closed.

CPExpert produces Rule CIC266 if the number of Failed Link Allocates (A20ESTAF) is greater than the **ALLOCF** guidance variable, for requests to allocate **specific** sessions with a modegroup.

The default specification for the ALLOCF guidance variable is **%LET ALLOCF = 0**, indicating that no failed allocation requests are acceptable.

Suggestion: CPExpert suggests that you examine the state of the connection that CICS is trying to allocate a session for the modegroup. Resolve any problem causing the **specific** ALLOCATE requests to fail.

You should check the CSMT log for the CICS statistics interval reported by Rule CIC266 for an indication of the problem with the connection for the modegroup described by the rule output.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: page 60 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC267: Insufficient sessions may have been defined

Finding: CPExpert believes that insufficient sessions may have been defined for the CICS region. Alternatively, the application system could be issuing ALLOCATE requests too often.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

The number of ALLOCATE requests which were returned by CICS to the transaction is reported as Failed Allocates Due to Sessions in Use (A14ESTAO). This includes requests to allocate **generic** sessions for APPC connections.

CPExpert detects a **potential** problem when the number of Failed Allocates Due to Sessions in Use (A14ESTAO) is greater than the **ALLOCQ** guidance variable in USOURCE(CICGUIDE).

The default specification is **ALLOCQ=1**, indicating that Rule CIC267 would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available for a generic ALLOCATE request. **This low default value is intended only to alert you to a potential problem, and is intended to make you aware of this analysis mechanism**.

For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations. For example, some installations always have a few ALLOCATE requests which cannot be satisfied and are returned to the transaction.

Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few unsatisfied requests, but are concerned about the situations when hundreds or thousands of requests are returned.

Suggestion: CPExpert suggests that you consider increasing the number of sessions allocated to the connection name identified as having too few sessions. For APPC connections, you should consider increasing the number of sessions allocated to the first modegroup for the connection name. Please refer to the above discussion to assess whether the number of sessions should be increased.

Additionally, for APPC connections, please refer to Rule CIC262 for a discussion of the considerations related to **generic** versus **specific** ALLOCATE requests.

Alternatively, you may wish to examine the transaction issuing the generic ALLOCATE requests. It is possible that the transaction is reissuing the ALLOCATE request without a suitable time delay allow a session to be freed, or the transaction does not use some other mechanism which might result in a session becoming available. The result from this reissuing the ALLOCATE request might simply be a looping between CICS and the transaction.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: pages 55-56 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.15.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC268: Insufficient sessions may have been defined for modegroup

Finding: CPExpert believes that insufficient sessions may have been defined for a modegroup in the CICS region. Alternatively, the application system could be issuing ALLOCATE requests too often.

Impact: This finding should normally have a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. Please refer to Rule CIC260 for a discussion of basic ISC/IRC concepts.

Discussion: Transactions acquire the use of a session in an ISC/IRC environment by using the ALLOCATE command. Conversations can take place between the two CICS regions or systems only after the session has been allocated. Once established, the session normally exists for a long time and can be used by many different transactions. The session normally is terminated by a FREE command.

A session must be available in order to be allocated in response to the ALLOCATE command. If a session is not available, CICS will normally queue the allocate request (and suspend the transaction) until a session is made available. Optionally (using the NOQUEUE specification), control can be returned to the transaction which can take application-dependent action based on the unavailability of a session.

The number of ALLOCATE requests which were returned by CICS to the transaction is reported as Failed Allocates Due to Sessions in Use (A20ESTAO) for specific ALLOCATE requests.

CPExpert detects a **potential** problem when the number of Failed Allocates Due to Sessions in Use (A20ESTAO) is greater than the **ALLOCQ** guidance variable in USOURCE(CICGUIDE).

The default specification is **ALLOCQ=1**, indicating that Rule CIC268 would be produced if more than one ALLOCATE request could not be satisfied because no sessions were available for a specific modegroup. **This low default value is intended only to alert you to a potential problem, and is intended to make you aware of this analysis mechanism**.

For many installations, the default should be changed after executing the CICS Component a few times. The **ALLOCQ** variable should normally be used to cause CPExpert to signal a problem only when you wish to be informed of abnormal situations.

For example, some installations always have a few specific ALLOCATE requests which cannot be satisfied and are returned to the transaction. Occasionally, however, several hundred or even several thousand requests are queued. Analysts at these installations are not concerned about the few unsatisfied requests, but are concerned about the situations when hundreds or thousands of requests are returned.

Suggestion: CPExpert suggests that you consider increasing the number of sessions allocated to the modegroup identified as having too few sessions. Please refer to the above discussion to assess whether the number of sessions should be increased.

Alternatively, you may wish to examine the transaction issuing the specific ALLOCATE requests. It is possible that the transaction is reissuing the ALLOCATE request without a suitable time delay allow a session to be freed, or the transaction does not use some other mechanism which might result in a session becoming available. The result from this reissuing the ALLOCATE request might simply be a looping between CICS and the transaction.

Reference: *CICS/ESA Version 3.1.1 Performance* Guide: pages 76-84.

CICS/ESA Version 3.2.1 Performance Guide: pages 294-301.

CICS/ESA Version 3.3.1 Performance Guide: pages 55-56 and pages 313-320.

CICS/ESA Version 4.1.1 Performance Guide: Section 2.2.23 and Appendix A.1.13.

CICS/TS Release 1.1 Performance Guide: Section 2.2.23 and Appendix 1.1.14.

CICS/TS Release 1.2 Performance Guide: Section 2.2.24 and Appendix 1.1.14.

CICS/TS Release 1.3 Performance Guide: Section 2.2.26 and Appendix 1.1.15.

CICS/TS Release 1.3 Performance Guide: Section 2.2.25 and Appendix 1.1.14.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 5 (ISC/IRC system and mode entry statistics) and Appendix A (Table 64).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.27 (Interpreting ISC/IRC system and mode entry statistics) and Appendix 1.1.12.

Rule CIC275: CICS-DB2 TCBs in use is approaching TCBLIMIT

Finding: The peak number of Task Control Blocks (TCBs) in use for the CICS-DB2 connection is approaching the limit set by the TCBLIMIT parameter.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of CICS tasks in the region that use the CICS-DB2 connection.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. This finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and subsequent releases of CICS.

Discussion: This rule was added to CPExpert Release 13.1 in April 2003 at the request of **Rex Avendano** (Kaiser Permenante), and Release 13.1 was issued at the end of April. The work involved with issuing Release 13.1 precluded any comprehensive research into the finding. Consequently, while the rule is produced by CPExpert code, no discussion of the implications or suggestions for alternative actions have been provided. This missing documentation will be produced immediately after Release 13.1 has been issued; I will have the documentation ready by mid-May. If any user has Rule CIC275 produced, please contact Don_Deese@cpexpert.com and I will send the new documentation to you.

CICS-DB2 global statistics are available in MXG file CICDB2GL. CPExpert uses data in CICDB2GL to calculate the percent of TCBs in use relative to the TCB Limit, using the following algorithm:

$$\textit{Percent TCBs in use} = \frac{\textit{Peak TCBs in use}}{\textit{TCB Limit}}$$

where

Peak TCBs in use	=	D2GTCBPK
TCB Limit	=	D2GTCBLM

CPExpert produces Rule CIC275 when the percent peak TCBs in use is more than the value specified by the **PCTD2TCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2TCB** is 80 indicating that CPExpert should produce Rule CIC275 whenever the peak TCBs in use was more than 80% of the TCB limit specified for the CICS-DB2 Connection.

Suggestion: Suggestions will be available upon request, in May 2003 when this document is revised.

Reference: References will be available upon request, in May 2003 when this document is revised.

Thanks: Thanks to **Rex Avendano** (Kaiser Permenante) for suggesting this rule.

Rule CIC276: High number of CICS-DB2 tasks on Pool Ready Queue

Finding: The peak number of tasks on the Pool Ready Queue is high for the CICS-DB2 connection.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of CICS tasks in the region that use the CICS-DB2 connection.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. This finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and subsequent releases of CICS.

Discussion: This rule was added to CPExpert Release 13.1 in April 2003 at the request of **Rex Avendano** (Kaiser Permenante), and Release 13.1 was issued at the end of April. The work involved with issuing Release 13.1 precluded any comprehensive research into the finding. Consequently, while the rule is produced by CPExpert code, no discussion of the implications or suggestions for alternative actions have been provided. This missing documentation will be produced immediately after Release 13.1 has been issued; I will have the documentation ready by mid-May. If any user has Rule CIC276 produced, please contact Don_Deese@cpexpert.com and I will send the new documentation to you.

CICS-DB2 global statistics are available in MXG file CICDB2GL. CPExpert produces Rule CIC276 when more than one task is on the Pool Ready Queue.

Suggestion: Suggestions will be available upon request, in May 2003 when this document is revised.

Reference: References will be available upon request, in May 2003 when this document is revised.

Thanks: Thanks to **Rex Avendano** (Kaiser Permenante) for suggesting this rule.

Rule CIC277: CICS-DB2 pool threads in use approaching thread limit

Finding: The number of pool threads in use is approaching the thread limit specified for the CICS-DB2 connection.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of CICS tasks in the region that use the CICS-DB2 connection.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics. This finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and subsequent releases of CICS.

Discussion: This rule was added to CPExpert Release 13.1 in April 2003 at the request of **Rex Avendano** (Kaiser Permenante), and Release 13.1 was issued at the end of April. The work involved with issuing Release 13.1 precluded any comprehensive research into the finding. Consequently, while the rule is produced by CPExpert code, no discussion of the implications or suggestions for alternative actions have been provided. This missing documentation will be produced immediately after Release 13.1 has been issued; I will have the documentation ready by mid-May. If any user has Rule CIC277 produced, please contact Don_Deese@cpexpert.com and I will send the new documentation to you.

CICS-DB2 global statistics are available in MXG file CICDB2GL. CPExpert uses data in CICDB2GL to calculate the percent of pool threads in use relative to the Pool Thread Limit, using the following algorithm:

$$\textit{Percent pool threads in use} = \frac{\textit{Peak pool threads in use}}{\textit{Pool Thread Limit}}$$

where

Peak pool threads in use	=	D2GTHRPK
Pool Thread Limit	=	D2GTHRLM

CPExpert produces Rule CIC277 when the percent pool threads in use is more than the value specified by the **PCTD2THR** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTD2THR** is 80 indicating that CPExpert should produce Rule CIC277 whenever the pool threads in use was more than 80% of the pool thread limit specified for the CICS-DB2 Connection.

Suggestion: Suggestions will be available upon request, in May 2003 when this document is revised.

Reference: References will be available upon request, in May 2003 when this document is revised.

Thanks: Thanks to **Rex Avendano** (Kaiser Permenante) for suggesting this rule.

Rule CIC301: CICS waited for logger "structure full" condition

Finding: The CICS interval statistics showed that CICS waited for logger "structure full" conditions.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

CICS uses the MVS system logger[1] for all logging and journaling requirements. Using services provided by the MVS system logger, the CICS log manager supports:

- The CICS **system log**, which is used for dynamic transaction backout, emergency restart, and preserving information for resynchronizing in-doubt units-of-work.

  Each CICS region has only one system log (the system log can be defined as a dummy if it is not required by the region). The CICS system log is intended for use only for recovery purposes (for example, it would be used during dynamic transaction backout, or during emergency restart).

  In CICS Transaction Server for OS/390, the system log is implemented as two MVS system logger log streams, but together they form a single logical stream. System log stream names are generally qualified names, where the last qualification is *DFHLOG* for the primary and *DFHSHUNT* for the secondary stream name.

- Forward recovery logs, auto-journals, and user journals. The CICS forward recovery logs, autojournals, and user journals are referred to as **general logs**, to distinguish them from *system logs*. This distinction in important when analyzing performance data. These logs are kept separate from the system log. CICS checks their stream names to ensure that they are different from that of the system log.

[1]The MVS system logger allows applications (e.g., CICS) to write records into a log stream. The MVS system logger provides sysplex-wide services and ensures data integrity for multiple systems accessing the log stream. 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 Release 2.4 (or Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

The CICS Log Manager uses the MVS system logger instead of CICS journal data sets. The CICS Log Manager provides online merging of general log streams from different CICS regions, which may be on different MVS images in a sysplex. The MVS system logger direct access to blocks of log data. This feature enables the CICS log manager to read the system log directly during dynamic transaction backout.

From the CICS 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 dataset, in a coupling facility, or in a log stream dataset. During installation of MVS, system parameters control the placement and length of log stream.

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.

The MVS system logger allows applications (e.g., CICS) to write records into a log stream. The MVS system logger provides sysplex-wide services and ensures data integrity for multiple systems accessing the log stream. 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 Release 2.4 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility. Beginning with CICS/TS for OS/390 Release 1.2, CICS can exploit the DASD-only or coupling facility options for individual log streams.

The MVS logger is not always able to satisfy immediately requests from the CICS Log Manager. This problem normally occurs in two situations

- 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[2]. This condition is analyzed by Rule CIC301.

- For a DASD-only log stream, 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 applies only with CICS/TS Release 1.2 and later releases, and is analyzed by Rule CIC302.

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

[2]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!

data. Both situations cause CICS to wait before it can write more data. Depending on the length of time CICS must wait, significant performance degradation would be experienced by the CICS region.

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 TYPE88 data set with the **STRFULL** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC301 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 CIC301 when any structure full condition was detected.

Suggestion: IBM suggests that you consider the following alternatives to reduce CICS waits for buffer full conditions:

- Increase the size of the coupling facility structure in order to smooth out spikes in logger load.

- Reduce the data written to the log stream by not merging so many journals or forward recovery logs onto the same stream.

- Reduce the HIGHOFFLOAD threshold percentage (the point at which the system logger begins off loading data from primary storage to off-load data sets).

- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. CPExpert normally will produce Rule CIC307 if too many DASD shifts occurred.

- Examine device I/O statistics for possible contention on the I/O subsystem used for off-load data sets.

- Use faster DASD devices.

Reference: *CICS/TS Release 1.1 Performance Guide*: Section 4.6.1 (Monitoring the logger environment).

CICS/TS Release 1.2 Performance Guide: Section 4.6.2 (Monitoring the logger environment).

CICS/TS Release 1.3 Performance Guide: Section 4.10.2 (Monitoring the logger environment).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22 (Monitoring the logger environment).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.9.1 Monitoring the logger environment)

Rule CIC302: CICS waited for logger "staging data set full" condition

Finding: The CICS interval statistics showed that CICS waited for logger "staging data set full" conditions.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics. The finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and later releases of CICS/TS.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

CICS uses the MVS system logger for all logging and journaling requirements. Using services provided by the MVS system logger, the CICS log manager supports:

- The CICS **system log**, which is used for dynamic transaction backout, emergency restart, and preserving information for resynchronizing in-doubt units-of-work.

  Each CICS region has only one system log (the system log can be defined as a dummy if it is not required by the region). The CICS system log is intended for use only for recovery purposes (for example, it would be used during dynamic transaction backout, or during emergency restart).

  In CICS Transaction Server for OS/390, the system log is implemented as two MVS system logger log streams, but together they form a single logical stream. System log stream names are generally qualified names, where the last qualification is *DFHLOG* for the primary and *DFHSHUNT* for the secondary stream name.

- Forward recovery logs, auto-journals, and user journals. The CICS forward recovery logs, autojournals, and user journals are referred to as **general logs**, to distinguish them from *system logs*. This distinction in important when analyzing performance data. These logs are kept separate from the system log. CICS checks their stream names to ensure that they are different from that of the system log.

The CICS Log Manager uses the MVS system logger instead of CICS journal data sets. The CICS Log Manager provides online merging of general log streams from different CICS regions, which may be on different MVS images in a sysplex. The MVS system logger direct access to blocks of log data. This feature enables the CICS log manager to read the system log directly during dynamic transaction backout.

From the CICS 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 dataset, in a coupling facility, or in a log stream dataset. During installation of MVS, system parameters control the placement and length of log stream.

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.

The MVS system logger allows applications (e.g., CICS) to write records into a log stream. The MVS system logger provides sysplex-wide services and ensures data integrity for multiple systems accessing the log stream. 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 Release 2.4 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility. Beginning with CICS/TS for OS/390 Release 1.2, CICS can exploit the DASD-only or coupling facility options for individual log streams.

The MVS logger is not always able to satisfy immediately requests from the CICS Log Manager. This problem normally occurs in two situations

- 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[1]. This condition is analyzed by Rule CIC301.

- For a DASD-only log stream, 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 applies only with CICS/TS Release 1.2 and later releases, and is analyzed by Rule CIC302.

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

[1] 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!

data. Both situations cause CICS to wait before it can write more data. Depending on the length of time CICS must wait, significant performance degradation would be experienced by the CICS region.

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(CICGUIDE). CPExpert produces Rule CIC302 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 CIC302 when any staging data set full condition was detected.

Suggestion: IBM suggests that you consider the following alternatives to reduce CICS waits for buffer full conditions:

- Increase the size of the staging data set in order to smooth out spikes in logger load.
- Reduce the data written to the log stream by not merging so many journals or forward recovery logs onto the same stream.
- Reduce the HIGHOFFLOAD threshold percentage, the point at which the system logger begins off loading data from primary storage to off-load data sets.
- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. Aim for no more than one DASD shift per hour. You can monitor the number of DASD shifts using the SMF88EDS record.
- Examine device IO statistics for possible contention on the I/O subsystem used for off-load data sets.
- Use faster DASD devices.

Reference: *CICS/TS Release 1.1 Performance Guide*: Section 4.6.1 (Monitoring the logger environment).

CICS/TS Release 1.2 Performance Guide: Section 4.6.2 (Monitoring the logger environment).

CICS/TS Release 1.3 Performance Guide: Section 4.10.2 (Monitoring the logger environment).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22 (Monitoring the logger environment).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.9.1 Monitoring the logger environment)

Rule CIC303: Log stream structure offloads occurred: 90% full

Finding: The SMF Type 88 data showed that log stream structure offloads occurred because the structure was 90% full. This finding applies only to log streams that are defined to use a coupling facility structure.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

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 contained in 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.

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[1] 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 **HIGHOFFLOAD 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 **LOWOFFLOAD threshold** is the point in the coupling facility structure, as a percent of space consumed, where the system logger stops offloading log stream data to DASD log 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.

From the above description, the amount of data that normally is offloaded is the difference between HIGHOFFLOAD and LOWOFFLOAD, as percentages of the coupling facility list structure size. For example, if the HIGHOFFLOAD value was specified as 80% and LOWOFFLOAD value was specified as 60%, 20% (80%-60%=20%) of the structure would be offloaded once offloading commenced.

The word "normally" has been used deliberately in the previous paragraphs. There are some situations when HIGHLOFFLOAD and LOWOFFLOAD parameters do not control the system logger offloading process.

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.

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

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%).

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the CICS interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

CPExpert examines the SMF88STN variable (the structure name) to select information that applies only to coupling facility structures[2].

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 CIC303 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.

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

This problem occurs primarily when more than one log stream uses a coupling facility structure and the applications using the log streams write a significantly different rates. Consequently, the offloads are being triggered by all the entries being used rather than triggered by the HIGHOFFLOAD value.

Suggestion: When Rule CIC303 is produced, you should consider the following alternatives:

- Review the log streams that share the coupling facility structure. IBM recommends that log streams sharing a coupling facility structure have similar rates of writing and similar amounts of data written.

- Review the size of the list structure in the coupling facility, to determine whether the structure size should be increased.

- You can alter CPExpert's analysis by modifying the STFULL90 guidance variable in USOURCE(CICGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*: Section 4.6.4 (LOWOFFLOAD and HIGHOFFLOAD parameters on log stream definition).

CICS/TS Release 1.2 Performance Guide: Section 4.6.5 (LOWOFFLOAD and HIGHOFFLOAD parameters on log stream definition).

CICS/TS Release 1.3 Performance Guide: Section 4.10.2 (Monitoring the logger environment).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22 (LOWOFFLOAD and HIGHOFFLOAD parameters on log stream definition).

OS/390 *MVS Setting up a Sysplex*:

OS/390 (V2R8): Section 9.4.3 (Determine the size of each coupling facility structure)
OS/390 (V2R9): Section 9.4.3 (Determine the size of each coupling facility structure)
OS/390 (V2R10): Section 9.4.3 (Determine the size of each coupling facility structure)

z/OS *MVS Setting up a Sysplex*:

Z/OS (V1R1): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R2): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R3): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R4): Section 9.4.3 (Determine the size of each coupling facility structure)

Rule CIC304: Interim storage was not efficiently used for log stream

Finding: The SMF Type 88 data showed that Interim storage (the coupling facility structure for the log stream) was not efficiently used for the log stream.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[1], 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[2] 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.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with

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

[2] DASD-only log streams are supported beginning with OS/390 Version 2 Release 4.

the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to 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 being filled.

For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure[3] 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 log stream data from the coupling facility list structure to DASD log data sets when 80% or more of the structure has been used.

- The **low offload threshold** is the point in the coupling facility structure, as a percent of 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.

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.

CICS manages 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.

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

[3] Please note that under certain conditions, a coupling facility log stream might be duplexed to staging data sets. If this should be the case, the HIGHOFFLOAD value applies to the staging data sets as well as to the coupling facility structure. See Rule CIC306 for additional information. For DASD-only log streams, duplexing to staging data sets is automatic.

TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the CICS interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

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 PCTINST with the **PCTINST** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC304 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTINST** guidance variable.

The default value for the **PCTINST** guidance variable is 0, indicating that CPExpert should produce Rule CIC304 whenever interim storage use was not effective.

Suggestion: The delete after offload percent is a key indicator that log tail deletion is not working as effectively as it should. If significant values appear in this percent, you should consider the following alternatives:

- Verify that SYSLOG=KEEP is not specified as a System Initialization Table (SIT) parameter (this suggestion applies only to CICS/TS Release 1.1, as the SYSLOG keyword was made obsolete with CICS/TS Release 1.2). The SYSLOG=KEEP option inhibits CICS from deleting data from the system log, even though the data is no longer needed. IBM strongly recommends that the SYSLOG=NOKEEP option be used, and the SYSLOG keyword was removed from the SIT with CICS/TS Release 1.2.

- Verify that there are not any long running transactions making recoverable updates without syncpointing.

- Consider increasing the HIGHOFFLOAD threshold value.

- Examine the System Initialization Table (SIT) values for this region, and determine whether AKPFREQ is zero or is too high. With a CICS/ESA 4.1 region (or earlier), the AKPFREQ parameter specifies the number of consecutive blocks written to the system log data set. However, with CICS/TS for OS/390, the AKPFREQ parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken.

  - If AKPFREQ=0, CICS cannot perform log tail deletion until shutdown, by which time the system log will have spilled to secondary storage. This situation would elongate shutdown and cause unnecessary overhead.

  - The AKPFREQ parameter has a significant impact on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. During AKP processing, the system logger deletes records that are no longer of interest to CICS and moves records to DFHSHUNT for those tasks that did write any log records within the last AKP interval.

  - In an MRO environment, the keypoint program uses an appreciable amount of CPU capacity in processing persisting units of work such as those relating to mirror transactions waiting to process an implicit forget. This is exacerbated when the AKPFREQ value is low. An optimum setting of AKPFREQ allows many of these persistent units of work to complete during normal transaction processing activity. This minimizes the CPU processing used by the keypoint program. IBM suggests that you exercise caution in reducing the value of AKPFREQ below the default value.

- Consider increasing the size of the coupling facility structure.

- You can alter CPExpert's analysis by modifying the **PCTINTST** guidance variable in USOURCE(CICGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*:
Section 4.6.1 (Monitoring the logger environment).
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.2 Performance Guide:
Section 4.6.2: Monitoring the logger environment.
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.3 Performance Guide:
Section 4.10.2: Monitoring the logger environment.
Section 4.10.7: Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.9.1 (Monitoring the logger environment) and Section 4.9.6 (Setting the activity keypoint frequency)

Rule CIC305: Staging data sets not efficiently used, DASD-only log stream

Finding: The SMF Type 88 data showed that staging data sets were not efficiently used for a DASD-only[1] log stream.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.2 and later releases of CICS/TS.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[2], 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.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with

[1]DASD-only log streams are supported beginning with OS/390 Version 2 Release 4.

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

the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to 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 being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[3].

For log streams defined as DASD-only, these parameters apply to the log stream staging data set, as follows:

- When the staging data set is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the staging data set 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 log stream data from the staging data set to DASD log data sets when 80% or more of the staging data set has been used.

- The **low offload threshold** is the point in the staging data set, as a percent of space consumed, where the system logger stops offloading log data in the staging data set 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.

CICS manages 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.

CICS collects statistics on the data written to each journal and log stream; this data can be used to analyze the activity of a single region. These statistics are available for analysis in the MXG CICLGR and CICLGS files, and CPExpert does analyze data from these files.

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

[3]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 CIC305.

TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the CICS interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

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

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(CICGUIDE). CPExpert produces Rule CIC304 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 CIC305 whenever DASD staging data set use was not effective.

Suggestion: The delete after offload percent is a key indicator that log tail deletion is not working as effectively as it should. If significant values appear in this percent, you should consider the following alternatives:

- Verify that SYSLOG=KEEP is not specified as a System Initialization Table (SIT) parameter (this suggestion applies only to CICS/TS Release 1.1, as the SYSLOG keyword was made obsolete with CICS/TS Release 1.2). The SYSLOG=KEEP option inhibits CICS from deleting data from the system log, even though the data is no longer needed. IBM strongly recommends that the SYSLOG=NOKEEP option be used, and the SYSLOG keyword was removed from the SIT with CICS/TS Release 1.2.

- Verify that there are not any long running transactions making recoverable updates without syncpointing

- Consider increasing the HIGHOFFLOAD threshold value.

- Examine the System Initialization Table (SIT) values for this region, and determine whether AKPFREQ is zero or is too high. With a CICS/ESA 4.1 region (or earlier), the AKPFREQ parameter specifies the number of consecutive blocks written to the system log data set. However, with CICS/TS for OS/390, the AKPFREQ parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken.

  - If AKPFREQ=0, CICS cannot perform log tail deletion until shutdown, by which time the system log will have spilled to secondary storage. This situation would elongate shutdown and cause unnecessary overhead.

  - The AKPFREQ parameter has a significant impact on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. During AKP processing, the system logger deletes records that are no longer of interest to CICS and moves records to DFHSHUNT for those tasks that did write any log records within the last AKP interval.

  - In an MRO environment, the keypoint program uses an appreciable amount of CPU capacity in processing persisting units of work such as those relating to mirror transactions waiting to process an implicit forget. This is exacerbated when the AKPFREQ value is low. An optimum setting of AKPFREQ allows many of these persistent units of work to complete during normal transaction processing activity. This minimizes the CPU processing used by the keypoint program. IBM suggests that you exercise caution in reducing the value of AKPFREQ below the default value.

- Consider increasing the size of the DASD staging data sets.

- You can alter CPExpert's analysis by modifying the **PCTLOCST** guidance variable in USOURCE(CICGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*:
Section 4.6.1 (Monitoring the logger environment).
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.2 Performance Guide:
Section 4.6.2: Monitoring the logger environment.
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.3 Performance Guide:
Section 4.8.2: Monitoring the logger environment.
Section 4.8.7: Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 4.9.1 (Monitoring the logger environment) and Section 4.9.6 (Setting the activity keypoint frequency)

Rule CIC306: DASD staging data set high threshold was reached

Finding: The SMF Type 88 data showed that the DASD staging data set high threshold was reached.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390. Additionally, this finding applies only to log streams that are defined to use a coupling facility.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

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*.

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.

From the above description, the amount of data that normally is offloaded is the difference between HIGHOFFLOAD and LOWOFFLOAD, as percentages of the coupling facility list structure size. For example, if the HIGHOFFLOAD value was specified as 80% and LOWOFFLOAD value was specified as 60%, 20% (80%-60%=20%) of the structure would be offloaded once offloading 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[1].

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the <u>CICS interval statistics</u> to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

CPExpert examines the SMF88STN variable (the structure name) in the MXG TYPE88 data set to select records that apply only to coupling facility structures[2]. 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 CIC306 when the SMF88ETF value exceeds the **STDSHIGH** guidance variable in USOURCE(CICGUIDE).

The default value for the **STDSHIGH** is zero, indicating that CPExpert should produce Rule CIC306 whenever a Staging Data Set Threshold was encountered.

[1] 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 CIC302.

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

Suggestion: IBM suggests that you size the staging data sets larger than the coupling facility structure size for the log streams.

While you can modify CPExpert's analysis by altering the STDSHIGH guidance variable, you should not do so unless you have unusual circumstances.

Reference: OS/390 *MVS Setting up a Sysplex*:

OS/390 (V2R8): Section 9.4.3 (Determine the size of each coupling facility structure)
OS/390 (V2R9): Section 9.4.3 (Determine the size of each coupling facility structure)
OS/390 (V2R10): Section 9.4.3 (Determine the size of each coupling facility structure)

z/OS *MVS Setting up a Sysplex*:

Z/OS (V1R1): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R2): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R3): Section 9.4.3 (Determine the size of each coupling facility structure)
Z/OS (V1R4): Section 9.4.3 (Determine the size of each coupling facility structure)

Rule CIC307: Frequent log stream DASD-shifts occurred

Finding: The SMF Type 88 data showed that frequent log stream DASD-shifts occurred.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

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.

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.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the CICS interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

CPExpert examines the SMF88EDS variable (the number of log stream DASD shifts during the SMF interval). Recall that IBM suggests that you not have more than one DASD shift per hour. However, 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 CIC307 when the number of DASD shifts that occur during any hour exceeds the **LGSHIFTS** guidance variable in USOURCE(CICGUIDE).

The default value for the **LGSHIFTS** is one, indicating that CPExpert should produce Rule CIC307 when more than one log stream DASD shift occurred during any hour.

Suggestion: If CPExpert produces Rule CIC307, you should consider the following alternatives:

- If more than one DASD shift occurs per hour, you should increase the size of the offload data sets. IBM recommends that you size the offload data sets as large as your installation can afford to make them. This will minimize the number of log data sets required to represent a log stream. It will also minimize the number of times that system logger must reallocate and switch to using a new log data set when an old one becomes full.

- You can alter CPExpert's analysis by changing the value of the **LGSHIFTS** guidance variable in USOURCE(CICGUIDE).

Reference: OS/390 *MVS Setting up a Sysplex*:

OS/390 (V2R8): Section 9.4.5 (Plan DASD Space for System Logger)
OS/390 (V2R9): Section 9.4.5 (Plan DASD Space for System Logger)
OS/390 (V2R10): Section 9.4.5 (Plan DASD Space for System Logger)

Z/OS *MVS Setting up a Sysplex*:

Z/OS (V1R1): Section 9.4.5 (Plan DASD Space for System Logger)
Z/OS (V1R2): Section 9.4.5 (Plan DASD Space for System Logger)
Z/OS(V1R3): Section 9.4.5 (Plan DASD Space for System Logger)
Z/OS(V1R4): Section 9.4.5 (Plan DASD Space for System Logger)

Finding: The SMF Type 88 data showed that the log stream caused its coupling facility structure to reach high threshold.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for OS/390.

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

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 contained in 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.

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[1] being filled.

[1] 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.

- When the interim storage (either coupling facility structure or staging data set) is filled to the **HIGHOFFLOAD threshold** point or beyond, the system logger begins offloading log data 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 **LOWOFFLOAD threshold** is the point in the interim storage (coupling facility structure or staging data set), as a percent of space consumed, where the system logger stops offloading log data to DASD log 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.

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 structure to DASD log data sets or if the system logger configuration is tuned incorrectly. The Type-3 completions are analyzed by Rule CIC309.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the <u>CICS</u>

interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

CPExpert examines the SMF88SC2 variable (Count of Type-2 completions) in the SMF Type 88 records. CPExpert produces Rule CIC308 when the SMF88SC2 value exceeds the **STRC2** guidance variable in USOURCE(CICGUIDE). The default value for the **STRC2** is zero, indicating that CPExpert should produce Rule CIC308 whenever the HIGHOFFLOAD threshold was reached in an SMF interval.

Suggestion: The number of Type-2 completions is simply a count of the number of times the HIGHOFFLOAD threshold for the coupling facility structure was reached based on writes to the specific log stream. Reaching the HIGHOFFLOAD threshold might or might not be an indication of a problem.

- You might wish log data to be frequently "hardened" to a DASD log data set. In this situation, you would define a relatively small coupling facility structure or specify a relatively low value for the HIGHOFFLOAD threshold. Consequently, you would expect to have Type-2 completions relatively often and a relatively large number of Type-2 completions would not be a cause for concern.

  If this condition applies to the log stream, you should consider "turning off" Rule CIC308 for this log stream. Please refer to Section 3 for instructions on how to "turn off" rules and for instructions on how to specify guidance for individual log streams or structures.

- You might have multiple log streams sharing the coupling facility structure, or you might not wish to experience the overhead of offloading. In this situation, a large number of Type-2 completions (with the corresponding overhead of offloading) might be cause for alarm.

  If this condition applies to the structure, you should consider separating the log streams that use the structure (either creating a new coupling facility structure or using a different distribution scheme for the log streams amongst the structures that are defined. As a general guidance, you should not have log streams with different characteristics sharing the same coupling facility structure. IBM recommends that log streams sharing a coupling facility structure have similar rates of writing and similar amounts of data written.

Reference: OS/390 MVS System Management Facilities

OS/390 (V2R4):	Section 9.1.1.2
OS/390 (V2R5):	Section 9.1.1.2
OS/390 (V2R6):	Section 9.1.1.2
OS/390 (V2R7):	Section 9.1.1.2
OS/390 (V2R8):	Section 9.1.1.2
OS/390 (V2R9):	Section 9.1.1.2
OS/390 (V2R10):	Section 9.1.1.2
z/OS (V1R1):	Section 9.1.1.2
z/OS (V1R2):	Section 9.1.1.2
z/OS (V1R3):	Section 9.1.1.2
z/OS (V1R4):	Section 9.1.1.2

Rule CIC309: Log stream consumed most of structure resources

Finding: The SMF Type 88 data showed that the log stream consumed most of its coupling facility structure resources.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only with CICS/Transaction Server for .

Discussion: The CICS Log Manager is a domain that was introduced with CICS/Transaction Server for OS/390. The CICS Log Manager replaces the journal control management function of earlier releases of CICS.

Please refer to Rule CIC301 for more general information about the interaction between CICS and the MVS system logger.

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 contained in 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.

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[1] being filled.

[1] 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.

- When the interim storage (either coupling facility structure or staging data set) is filled to the **HIGHOFFLOAD threshold** point or beyond, the system logger begins offloading log data 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 **LOWOFFLOAD threshold** is the point in the interim storage (coupling facility structure or staging data set), as a percent of space consumed, where the system logger stops offloading log data to DASD log 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.

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 number of Type-2 completions is simply a count of the number of times the HIGHOFFLOAD threshold for the coupling facility structure was reached based on writes to the specific log stream. Reaching the HIGHOFFLOAD threshold might or might not be an indication of a problem.

  - You might wish log data to be frequently “hardened” to a DASD log data set. In this situation, you would define a relatively small coupling facility structure or specify a relatively low value for the HIGHOFFLOAD threshold. Consequently, you would expect to have Type-2 completions relatively often and a relatively large number of Type-2 completions would not be a cause for concern.

- You might have multiple log streams sharing the coupling facility structure, or you might not wish to experience the overhead of offloading. In this situation, a large number of Type-2 completions (with the corresponding overhead of offloading) might be cause for alarm.

- 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.

  For example, the system logger's access to its DASD log data sets would be slowed if those data sets reside on the same device as some other heavily-used data sets.

  A Type-3 can also occur if many log streams are defined to share the same structure, because each newly defined log stream causes the system logger to dynamically repartition storage among the existing log streams.

  If a log stream has a large proportion of Type-3 completions, the system logger is getting dangerously close to the STRUCTURE FULL condition.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. Since the SMF Type 88 records are from a system view, the records do not contain information related to individual CICS regions.

The SMF Type 88 records do identify the structures and log streams to which the information applies. Consequently, CPExpert can use the CICS interval statistics to identify specific structures and log streams that apply to specific CICS regions. CPExpert can then select information from the SMF Type 88 records that describe the structures and log streams used by the particular CICS region.

CPExpert examines the SMF88SC3 variable (Count of Type-3 completions) in the SMF Type 88 records. CPExpert produces Rule CIC308 when the SMF88SC3 value exceeds the **STRC3** guidance variable in USOURCE(CICGUIDE). The default value for the **STRC3** is zero, indicating that CPExpert should produce Rule CIC309 whenever the space used by a log caused the coupling facility structure to reach a critical amount.

Suggestion: If this finding is produced, determine whether there was a failure that caused the system logger to be unable to promptly offload data. If a failure did occur, you probably should ignore this finding. If a failure was not experienced, you should consider the following alternatives:

- Determine whether the system logger configuration is tuned incorrectly. The system logger might be unable to offload data promptly if the DASD log data sets experience I/O contention with other systems data sets.

- Review the structure size, to ensure that the structure is adequately sized for the log stream(s) using the structure.

- Review the number of log streams assigned to the coupling facility structure. The system logger might not be able to respond adequately if too many log streams are defined to share the same structure.

- Examine the application responsible for the log stream activity to determine whether its use of the log stream has increased, and whether this increase is expected.

- Review the HIGHOFFLOAD and LOWOFFLOAD parameters for the log stream to determine whether these should be adjusted. If either parameter value is too large, the system logger might not be able to respond adequately. The system logger might not have time to offload sufficient log stream data when the HIGHOFFLOAD parameter value is reached, before the log stream uses most of the structure. The system logger might offload only a relatively small amount of data once offloading commences, if the LOWOFFLOAD parameter is too high. Either of these situations could indicate that the parameters are too large, or could simply be the result of the coupling facility structure being too small.

- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. Rule CIC307 would be produced by CPExpert if too many DASD shifts occurred. However, you might have altered the guidance to CPExpert for Rule CIC307. In this case, Rule CIC307 might not be produced even though DASD shifts could have delayed the offloading of the log stream(s) assigned to the coupling facility structure.

Reference: OS/390 MVS System Management Facilities

OS/390 (V2R8):	Section 9.1.1.2
OS/390 (V2R9):	Section 9.1.1.2
OS/390 (V2R10):	Section 9.1.1.2
z/OS (V1R1):	Section 9.1.1.2
z/OS (V1R2):	Section 9.1.1.2
z/OS (V1R3):	Section 9.1.1.2
z/OS (V1R4):	Section 9.1.1.2

Finding: CPExpert detected that the paging rate was too high for the CICS region.

Impact: This finding should normally have a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based upon an analysis of the CICS statistics.

Discussion: Page-in operations occur when CICS attempts to reference a virtual page and the page is not in central storage. A page fault occurs and MVS is required to bring the page into central storage. During this page-in operation, the CICS region stops processing until MVS fetches the page.

The time the CICS region spends waiting for a page is not normally serious if the page is retrieved from expanded storage. The time to fetch a page from expanded storage is very small (on the order of 40 - 75 microseconds).

The wait time is much more lengthy (typically on the order of 20-50 milliseconds) if the page must be fetched from auxiliary storage. During this more lengthy time, the entire CICS region stops processing. Note that this situation is unlike a TSO environment, in which only a single TSO user is delayed because of a page fault. With CICS, the entire region is delayed. This is because each TSO user is an "address space" from the MVS perspective, while the entire CICS region is considered an "address space" by MVS.

The time in which CICS stops processing delays the transaction experiencing the page fault. This delay might not be so significant unless the task has many page faults. However, all of the CICS region stops processing, so all other active tasks can be delayed. Since all tasks are delayed, the tasks remain in the region longer and the tasks retain the allocated resources longer. Therefore, in addition to the delay associated with the page fault resolution, serious paging can lead to CICS stress conditions as tasks are delayed.

Paging information is not available in CICS statistics. This information is available only if you have monitor data in a MXG or MICS performance data base.

The amount of paging which can be tolerated by a CICS region depends upon how many active tasks are normally in the region, depends upon the response objectives for the applications in the region, etc. Consequently, no single "good" or "bad" paging rate can be established. IBM's CICS Performance Guides suggest that less than one page per second is best, less than five pages per second might be acceptable, and more than ten pages per second is a major problem. These values must be evaluated considering the probability that any average page rate reflects periods in which the page rate is many times the average.

CPExpert fires Rule CIC310 if the average page-in rate (pages per second) was greater than the **CICPAGIN** guidance variable. The default specification for this variable is **%LET CICPAGIN = 2**, indicating that the CICS region should not experience more than an average two pages per second. You can provide different guidance to CPExpert by changing the CICPAGIN variable if you feel that Rule CIC310 is firing prematurely.

Suggestion: There are many ways to reduce the paging rate for CICS, and the "best" way varies from installation to installation. CPExpert suggests that you consider the following alternatives to reduce the page-in rate:

- **Implement storage isolation for the CICS region**. This alternative is the easiest, and is the most common. Storage isolation is accomplished by using the PWSS parameter in IEAIPSxx for the performance group to which the CICS region is assigned. Implementing storage isolation would guarantee a minimum target working set to the CICS region.

  You normally should set the minimum protected working set (the first value in PWSS parameter) based upon the average working set size. You can obtain the average working set size from inspecting RMF Monitor II display (Address Space State Date Report).

- **Dynamically control the target working set for CICS region**. PPGRTR is a keyword in the IEAIPSxx member of SYS1.PARMLIB that can be used to dynamically control both the amount of paging and the central storage used by the private area.

  If PPGRTR is not specified, the target (or protected) working set is always at the minimum value specified for the PWSS keyword. If PPGRTR **is** specified, the target (or protected) working set is dynamically adjusted based upon the page-in rate. The page-in rate is calculated every 10 seconds. The page-in rate is based upon the number of page-in operations from auxiliary storage per second of residency time.

  When the page-in rate from auxiliary storage falls below the value specified as the first parameter in the PPGRTR keyword, the target working set is decreased. The reason for this action is that paging has fallen to a level such that sufficient private pages are in memory to satisfy memory demands. The implication is that memory demand does not require the target working set to be as large, and too much is being protected. The System Resources Manager (SRM) reduces the target working set by 3% when the paging rate falls below the value of the first parameter.

When the page-in rate from auxiliary storage rises above the value specified as the second parameter in the PPGRTR keyword, the target working set is increased. The reason for this action is that paging has risen to a level such that sufficient pages are **not** in memory to satisfy memory demands. The implication is that memory demand requires the target working set to be larger to reduce page replacement. The SRM increases the target working set by 7% when the paging rate rises above the value of the second parameter.

Thus, the PPGRTR keyword allows the target working set size to be dynamically adjusted, based upon varying memory demands. Central storage is protected when needed to minimize paging, but made available when not needed. Please refer to Rule SRM113 in the SRM Component User Manual for further discussion of the PPGRTR keyword.

For example, specify **PPGRTR=(2,5)** to indicate that the target working set should be increased when the paging rate rises above 5 pages per second and that the target working set should be decreased when the paging rate falls below 2 pages per second.

- **Specify "negative" storage isolation for selected low priority workloads**. When storage isolation is specified, the page stealing algorithm does a "pre-scan" of address spaces, stealing pages from any address spaces whose working set exceeds the maximum protected working set (the maximum value specified in PWSS). (Actually, it also includes the maximum value specified in CWSS, but specifying negative storage isolation for the common area is strongly discouraged.)

  Thus, page stealing can be directed to specific workloads by specifying a maximum value in PWSS that is less than the amount of storage commonly used by these workloads. "Negative" storage isolation might, for example, be used with low priority batch jobs to allow more page frames for use by CICS.

- **Limit the number of competing workloads requiring processor storage**. This can be done by (1) controlling the system multiprogramming level based upon paging rate (adjust the

RCCPTRT parameter in the IEAOPTxx member), (2) controlling the system multiprogramming level based upon page delay time (adjust the RCCPDLT parameter in the IEAOPTxx member), (3) controlling the system multiprogramming level based upon real memory use (adjusting the RCCUICT parameter in the IEAOPTxx member), or (4) limiting the multiprogramming level of selected workloads (reduce the maximum MPL of the CNSTR parameter of selected domains in IEAIPSxx). (Please refer to RULE SRM205 in the SRM Component's User Manual for a discussion of MPL controls.)

- **Workload scheduling**. Schedule lower priority workloads to a time when they do not compete with CICS.

- **Decrease the amount of central storage used for logical swapping** (adjust the LSCTxxx parameters in IEAOPTxx). If less central storage is used for logical swapping, more central storage will be available for active working sets and less paging will occur. Note, however, that decreasing logical swapping will increase the swap-in delay. This delay could have a detrimental effect on TSO performance. (Please refer to RULE SRM202 in the SRM Component's User Manual for a discussion of logical swapping controls.)

If the above alternatives can not be used to reduce the paging for the CICS region, you should consider alternatives to improve the page fault resolution.

- **Place paging devices on dedicated paths.** Significant performance improvement can be realized by placing paging devices on dedicated paths. Consider reconfiguring to place the paging devices on dedicated paths, if they are not already on dedicated paths.

- **Ensure that local page data sets reside on dedicated (or low utilized) devices**. Considerable queuing delays may result if local page data sets reside on devices which have other active data sets. RULE MVS140 will fire if this is a likely problem. (RULE MVS140 analyzes the number of I/O operations to local page data sets, compares that number with the number of page I/O operations to the data set, and fires if the difference is more than a token amount.)

- **Increase the number of local page data sets.** Significant performance improvements might be realized by increasing the number of local page data sets. This will allow more paths to be used, and less queuing to result. Perhaps more importantly, the page slots will be placed on more physical devices and there is less chance of seek delay.

- **Acquire faster paging devices**. If the above options have been exhausted and paging delays are still unacceptable, you should consider acquiring faster paging devices.

Additionally, the CICS Performance Guides for Version 1.7 and Version 2.1.2 list many application-related changes which can be implemented to reduce paging effects.

Reference: *CICS/OS/VS Version 1.7 Performance* Guide: pages 158-159 and pages 374-376.

CICS/MVS Version 2.1.2 Performance Guide: pages 89-91 and pages 335-336.

CICS/ESA Version 3.1.1 Performance Guide: pages 175-176.

CICS/ESA Version 3.2.1 Performance Guide: pages 79-80.

CICS/ESA Version 3.3.1 Performance Guide: pages 89-90.

CICS/ESA Version 4.1.1 Performance Guide: Section 3.2.5.

CICS/TS Release 1.1 Performance Guide: Section 3.2.5.

CICS/TS Release 1.2 Performance Guide: Section 3.2.5.

CICS/TS Release 1.3 Performance Guide: Section 3.2.5.

Rule CIC320: A large percent of structure list entries were in use

Finding: The CICS Shared Temporary Storage (TS) Queue Server statistics showed that a large percent of structure list entries had been used in the coupling facility structure containing the shared TS pool.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage (TS) are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. These queues are related to a shared TS pool using the SYSID keyword to specify the shared queue pool to which the request is directed.

Each TS pool is defined, using MVS cross-system extended services (XES), as a keyed list structure in a coupling facility.

Access to a TS pool by CICS transactions running in an AOR is through a *TS data sharing server* that supports a specific named pool. A shared TS pool server is started in an MVS image by starting a queue server region (as either a batch job or a started task) for each shared TS pool. Starting the queue server region invokes the queue server region program, DFHXQMN, which resides in an APF-authorized library. Each TS server provides access to only one pool of TS queues, so there must be multiple TS server regions if there are multiple shared TS pools.

A variety of parameters are provided to the TS pool server, to specify the name of the shared TS pool, list structure attributes, tuning parameters, various thresholds for warning and automatic server actions, etc. These parameters (other than the pool name) have default values, but the defaults can be changed based on user-specific requirements.

The application tasks issue a few general commands to read, write, or delete shared TS storage (READQ TS, WRITEQ TS, and DELETEQ TS). The queue server region is responsible for translating these application task commands into XES list structure interfaces.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With shared TS queues, the list structure is the named pool, while the lists themselves are shared TS queues within the named pool. The shared TS pool server designates the maximum number of lists (or queues) the TS pool is to have, and allocates the list structure based on parameters that are provided to the TS pool server.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both.

- The first connector to a list structure specifies whether the list structure has adjunct areas, and that status (presence or absence of an adjunct area) is fixed for the life of the structure. An adjunct area can be used to hold up to 64 bytes of data. With shared TS pools, this area contains the read cursor for small queues and the queue status information.

- Data entries are composed of units of storage called *data elements*. In a coupling facility of CFLEVEL=0, data entries can be composed of 0 to 16 data elements. In a coupling facility of CFLEVEL=1 or higher, data entries can be composed of 0 to 255 data elements. In either case, a data entry could contain up to 64K (65536 bytes) of data.

  While data elements can be 256, 512, 1024, 2048, or 4096 bytes in size, the default value for shared TS pools is 256 bytes. IBM states that "there is no known reason to specify other than the default value of 256."

  Although a data entry can be composed of a number of data elements, list operations treat the data entry as a single entity; data elements cannot be read or written individually.

When the list structure is allocated, XES establishes the number of data elements that are associated with data entries. This division of storage is is referred to as the "entry-to-element ratio".

The structure *alter* function provides for the expansion or contraction of the size of a structure, the reapportionment of the entry-to-element ratio of the structure's storage, and the alteration of the percentage of structure storage set aside for event monitor controls (EMCs). The structure alter processing is done either by using the IXLALTER macro or by issuing the SETXCF START,ALTER command. The IXLALTER macro allows an authorized user to request a change to the structure's size, the entry-to-element ratio, and the percentage of storage allocated for EMCs.

Starting with OS/390 Release 10, a structure can be *automatically* altered when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both.

With shared temporary storage, the queue server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued. The warning message is repeated each time the number in use increases beyond further thresholds.

Each time the warning is issued, the TS queue server tests whether an automatic ALTER for the entry to element ratio should be performed. The test is done based on the TS queue server *automatic structure alter parameters*. This is based on the ratio between the current numbers of elements and entries actually in use.

IBM suggests that no more than 75% of the structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

Section 2.2.2.1 (Approximate storage calculations) of the CICS/TS *System Definition Guide* provides calculations that can be used to calculate initial sizing of the storage. However, this algorithm might not yield an adequate structure size for some environments.

Further, the default element to entry ratio in the shared TS server *tuning parameters* is a simple 1:1, which might not be optimum for any particular shared temporary storage environment.

Consequently, CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the maximum percent of the structure list entries that had been used, using the following algorithm:

$$\text{Maximum percent structure list entries used} = \frac{S1ENTRHI}{S1ENTRMX}$$

where S1ENTRHI = Maximum number of list entries used since last reset
S1ENTRMX = Total list entries in the currently allocated structure

CPExpert produces Rule CIC320 when the maximum percent structure list entries used is more than the value specified by the **TSPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTENT** is 70 indicating that CPExpert should produce Rule CIC320 whenever more than 70% of the list entries had been used.

Suggestion: Rule CIC320 indicates that the TS queue server either is exercising automatic alter algorithms, or is likely to exercise these algorithms as the thresholds in the *automatic structure alter parameters* are reached. If this finding is produced often, you should consider the following alternatives:

- Increase the initial amount of structure space that is available for the shared TS pool identified by this finding. Increasing the amount of initial structure space can be accomplished by increasing the INITSIZE (so more structure space is initially available). If more structure space is initially available, more entries and elements will be available and there is less probability that there will be a shortage of list entries.

- Increase the amount of storage allocated for the *maximum size* specified in the coupling facility resource management policy for the shared temporary storage queue identified by this finding. This action normally should be taken only if the structure size has approached the maximum size specified. Be aware that Rule CIC325 will be produced if the structure runs out of space, but frequent occurrence of Rule CIC320 might indicate a pending “no space” condition.

- You could change the TSPCTENT guidance variable in USOURCE(CICGUIDE) so Rule CIC320 is produced less often. This action is not recommended, however since you should be aware of the potential problems (it is particularly important to be aware of <u>pending</u> problems) revealed by Rule CIC320

.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC321: A large percent of structure data elements were in use

Finding: The CICS Shared Temporary Storage (TS) Queue Server statistics showed that a large percent of structure data elements had been used in the coupling facility structure containing the shared TS pool.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

Please refer to Rule CIC320 for a discussion of shared TS pools.

The structure *alter* function provides for the expansion or contraction of the size of a structure, the reapportionment of the entry-to-element ratio of the structure's storage, and the alteration of the percentage of structure storage set aside for event monitor controls (EMCs). The structure alter processing is done either by using the IXLALTER macro or by issuing the SETXCF START,ALTER command. The IXLALTER macro allows an authorized user to request a change to the structure's size, the entry-to-element ratio, and the percentage of storage allocated for EMCs.

Starting with OS/390 Release 10, a structure can be automatically altered when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both.

With Shared Temporary Storage, the queue server monitors the total number of entries and elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the TS queue server *warning parameters*, a warning message (DFHXQ0411 or DFHXQ0412, for entries and elements, respectively) is issued. The warning message is repeated each time the number in use increases beyond further thresholds.

Each time the warning is issued, the TS queue server tests whether an automatic ALTER for the entry to element ratio should be performed. The test is done based on the TS queue server *automatic structure alter parameters*. This is based on the ratio between the current numbers of elements and entries actually in use.

IBM suggests that no more than 75% of the structure be used to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

Section 2.2.2.1 (Approximate storage calculations) of the CICS/TS *System Definition Guide* provides calculations that can be used to calculate initial sizing of the storage. However, this algorithm might not yield an adequate structure size for some environments.

Further, the default element to entry ratio in the shared TS server *tuning parameters* is a simple 1:1, which might not be optimum for your shared temporary storage environment.

Consequently, CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the maximum percent of the data elements that had been used, using the following algorithm:

$$\mathit{Maximum\ percent\ data\ elements\ used} = \frac{\mathit{S1ELEMHI}}{\mathit{S1ELEMMX}}$$

where S1ELEMHI = Maximum number of data elements used since last reset
S1ELEMMX = Total data elements in the currently allocated structure

CPExpert produces Rule CIC321 when the maximum percent data elements used is more than the value specified by the **TSPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTELE** is 70 indicating that CPExpert should produce Rule CIC321 whenever more than 70% of the data elements had been used.

Suggestion: Rule CIC321 indicates that the TS queue server either is exercising automatic alter algorithms, or is likely to exercise these algorithms as the thresholds in the *automatic structure alter parameters* are reached. If this finding is produced often, you should consider the following alternatives:

- Increase the initial amount of structure space that is available for the shared TS pool identified by this finding. Increasing the amount of initial structure space can be accomplished by increasing the INITSIZE (so more structure space is initially available). If more structure space is initially available, more entries and elements will be available and there is less probability that there will be a shortage of data elements.

- Increase the amount of storage allocated for the *maximum size* specified in the coupling facility resource management policy for the shared temporary storage queue identified by this finding. This action normally should be taken only if the structure size has approached the maximum size specified. Be aware that Rule CIC325 will be produced if the structure runs out of space, but frequent occurrence of Rule CIC321 might indicate a pending "no space" condition.

- You could change the TSPCTELE guidance variable in USOURCE(CICGUIDE) so Rule CIC321 is produced less often. This action is not recommended, however since you should be aware of the potential problems (it is particularly important to be aware of pending problems) revealed by Rule CIC321.

.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC322: A large percent of structure requests were timed out by CF

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of structure requests were timed out by the coupling facility.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

Some list structure commands can complete prematurely, because the request exceeded the coupling facility model-dependent time-out criteria (Return and Reason Code equate symbol IXLRSNCODETIMEOUT). These list structure commands are:

IXLLIST REQUEST=DELETE_ENTRYLIST
IXLLIST REQUEST=DELETE_MULT
IXLLIST REQUEST=DEQ_EVENTQ
IXLLIST REQUEST=LOCK
IXLLIST REQUEST=MONITOR_EVENTQ
IXLLIST REQUEST=MONITOR_SUBLISTS
IXLLIST REQUEST=READ_LIST
IXLLIST REQUEST=READ_MULT
IXLLIST REQUEST=WRITE
IXLLSTC -- XES List Structure Control Services
IXLLSTE -- XES List Structure Single Entry Services
IXLLSTM -- XES List Structure Multiple Entry Services

When list structure commands complete prematurely, the application typically restarts the command (using information has been returned in the answer area). Commands that complete prematurely cause unnecessary overhead in the coupling facility and in the application.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of requests that timed out, using the following algorithm:

$$\textit{Percent requests that timed out} = \frac{\textit{S1RSP2CT}}{\textit{S1RSP1CT} + \textit{Abnormal responses}}$$

where S1RSP1CT = Number of normal responses
S1RSP2CT = Number of requests that timed out
Abnormal = S1RSP2CT + S1RSP3CT + S1RSP4CT + S1RSP5CT+ S1RSP6CT + S1RSP7CT + S1RSP8CT

S1RSP3CT = Specified entry not found
S1RSP4CT = Version check failed for entry being updated
S1RSP5CT = List authority comparison failed
S1RSP6CT = Maximum list key reached
S1RSP7CT = List structure was out of space
S1RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC322 when the percent requests that timed out is greater than the value specified by the **TSPCTTIM** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTTIM** is 1 indicating that CPExpert should produce Rule CIC322 whenever more than one percent of the requests timed out because the request exceeded the coupling facility model-dependent time-out criteria.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of shared temporary storage pools to determine whether the IXLLIST commands can be issued with fewer actions against temporary storage entries. For example, the DELETEQ TS command might be issued more frequently to delete a smaller number of entries with each delete.

- Change the TSPCTTIM guidance variable in USOURCE(CICGUIDE) so Rule CIC322 is produced only when you wish to be aware of applications that cause a larger percent of coupling facility time-outs.

- You can specify **%LET TSPCTTIM = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent requests timed-out cannot be greater than 100), or you can "turn off" the rule using the process described in Section 3 of this User Manual.

Reference: z/OS V1R1 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R2 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R3 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R4 *MVS Programming: Sysplex Services Reference*
Section 46.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

Rule CIC323: High percent entries (either queue or item) not found

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of entries (either queue or item) were not found.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct_access storage device, or in a temporary storage data sharing pool. Several commands can be used to reference, modify, or delete data in a temporary storage queue or to delete the entire queue.

- Write data to a temporary storage queue (WRITEQ TS)
- Update data in a temporary storage queue (WRITEQ TS REWRITE)
- Read data from a temporary storage queue (READQ TS command)
- Read the next data from a temporary storage queue (READQ TS NEXT)
- Delete a temporary storage queue (DELETEQ TS command)

When any of the above commands are executed, the response from the command can be “normal” in that no further action is required, or the response can be “abnormal” in that either the task is terminated or action is required. A *HANDLE CONDITION* command can be issued before the temporary storage commands are issued. The *HANDLE CONDITION* command can specify the label to which control is to be passed if a condition occurs.

Two abnormal conditions that apply to shared temporary storage are reported in the CICS statistics: *ITEMERR* and *QIDERR*.

- The ITEMERR abnormal condition can occur with the READQ TS command and with the WRITEQ TS command.

- The ITEMERR abnormal condition can occur with the READQ TS command when (1) the item number specified is invalid (that is, the item number is outside the range of item numbers written to the queue), or (2) an attempt is made to read beyond the end of the queue using the NEXT (default) option.

  A temporary storage queue that has only one record can be treated as a single unit of data that can be accessed using its symbolic name. Using temporary storage control in this way provides a typical scratch-pad capability. This type of storage should be accessed using the READQ TS command with the ITEM option; not doing so may cause the ITEMERR condition to be raised.

- The ITEMERR abnormal condition can occur with the WRITEQ TS command when (1) the item number specified in a WRITEQ TS command with the REWRITE option[1], is not valid (that is, it is outside the range of entry numbers assigned for the queue), or (2) the maximum number of items (32 767) is exceeded.

- The QIDERR abnormal condition can occur with the READQ TS command, with the WRITEQ TS command, and with the DELETEQ TS command. The QIDERR abnormal condition occurs when the specified queue cannot be found. The default action (if the *HANDLE CONDITION* command has not been issued) is to terminate the task abnormally.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a "specified entry (queue or item) was not found" condition, using the following algorithm:

$$Percent\ entries\ not\ found = \frac{S1RSP3CT}{S1RSP1CT + Abnormal\ responses}$$

where S1RSP1CT = Number of normal responses
Abnormal = S1RSP2CT + S1RSP3CT + S1RSP4CT + S1RSP5CT+ S1RSP6CT + S1RSP7CT + S1RSP8CT

S1RSP2CT = Requests timed out by CF
S1RSP3CT = Specified entry not found
S1RSP4CT = Version check failed for entry being updated
S1RSP5CT = List authority comparison failed

[1] If the REWRITE option is specified, the ITEM option must also be specified. If the specified queue does not exist, the QIDERR condition occurs. If the correct item within an existing queue cannot be found, the ITEMERR condition occurs and the data is not stored.

S1RSP6CT = Maximum list key reached
S1RSP7CT = List structure was out of space
S1RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC323 when the percent requests that encountered a “specified entry (queue or item) was not found” condition is greater than the value specified by the **TSPCTNOE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTNOE** is 0.1 indicating that CPExpert should produce Rule CIC323 whenever more than one tenth percent of the requests encountered a “specified entry (queue or item) was not found” condition.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Examine the application to determine why the “specified entry (queue or item) was not found” condition is encountered. Under normal situations, this condition should be considered a program error and should be remedied.

- Change the TSPCTNOE guidance variable in USOURCE(CICGUIDE) so Rule CIC323 is produced only when you wish to be aware of applications that cause a larger percent of “specified entry (queue or item) was not found” condition.

- You can specify **%LET TSPCTNOE = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent specified entry (queue or item) cannot be greater than 100), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS Application Programming Reference:
Section 1.60 DELETEQ TS
Section 1.140 READQ TS
Section 1.241 WRITEQ TS

CICS/TS for OS/390 Release 1.2
CICS Application Programming Reference:
Section 1.60 DELETEQ TS
Section 1.140 READQ TS
Section 1.241 WRITEQ TS

CICS/TS for OS/390 Release 1.3
CICS Application Programming Reference:
Section 1.62 DELETEQ TS

Section 1.150 READQ TS
Section 1.268 WRITEQ TS

CICS/TS for z/OS Release 2.1
CICS Application Programming Reference:
CICS API commands (READQ TS, WRITEQ TS, and DELETEQ TS)

CICS/TS for z/OS Release 2.2
CICS Application Programming Reference:
CICS API commands (READQ TS and WRITEQ TS)

Rule CIC324: Maximum key list was reached for shared temporary storage

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that the maximum list key had been reached for shared temporary storage, indicating that the maximum queue size or the maximum queues had been reached (depending on the list).

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region. Three parameters control how many shared temporary storage queues can be in a specific pool and the characteristics of the queues:

- The *MAXQUEUES* parameter specifies the maximum number of data lists to be reserved when the structure is allocated. This parameter determines the maximum number of large queues that can be stored in the structure. The default value for the MAXQUEUES parameter is 1000, indicating that a maximum of 1000 large queues can be stored in the structure.

- The *SMALLQUEUEITEMS* specifies the maximum number of items that can be stored in the small queue format in the queue index entry data area. This parameter can force a queue to be converted to the large queue format if it has a large number of small items. This conversion "uses" one of the large queues that was specified by the MAXQUEUES parameter, in the sense that it adds to the current number of large queues stored in the structure.

  The default value for the *SMALLQUEUEITEMS* parameter is 9999, indicating that a maximum of 9999 items can be stored in the small queue format in the queue index entry data area before the queue would be converted to the large queue format.

- The *SMALLQUEUESIZE* parameter specifies the maximum data size for a small queue including the two-byte length prefix on each data item. Any queue exceeding the maximum size, when writing the second or subsequent item to a queue, is converted to the large queue format. This conversion "uses" one of the large queues that was specified by the MAXQUEUES parameter, in the sense that it adds to the current number of large queues stored in the structure.

The default value for the *SMALLQUEUESIZE* parameter is 32K, indicating that the queue would be converted to the large queue format when the queue size was more than 32K.

If the number of queues is reached (as set by the MAXQUEUES server initialization parameter described above), any further request to establish a queue will fail, and message **DFHXQ0443** (*CF structure strname request failed, all lists are in use*) will be issued. The failing request is given a NOSPACE indication if it originated from a CICS API request.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to determine whether a List Full condition occurred. CICXQ1 variable S1RSP6CT (List full: maximum list key reached) indicates that maximum queue size or maximum queues were reached, depending on the list.

CPExpert produces Rule CIC324 when the number of List Full conditions is greater than the value specified by the **TSLSTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSLSTFUL** is 0, indicating that CPExpert should produce Rule CIC324 whenever any List Full conditions occurred.

Suggestion: If this finding is produced, you should examine the output from Rule CIC324 to determine whether the highest number of queues in the TS pool is as large as the MAXQUEUES parameter.

- If the highest number of queues in the TS pool is **equal to** the MAXQUEUES parameter, the finding was produced because the maximum number of queues had been reached. In this case, you should consider the following alternatives:

  - Review the application to ensure that any queues of total size greater than 32K bytes are deleted when they are no longer in use. This action will free up data lists.

  - The number of lists is fixed when the structure is allocated. Consequently, the only way to increase the number of lists in a structure is to (1) unload the structure, (2) use SETXCF FORCE to delete the structure, and (3) reload the structure with a larger MAXQUEUES parameter.

- If the highest number of queues in the TS pool is **less than** the MAXQUEUES parameter, the finding was produced because the maximum data size for a small queue had been reached, or the maximum number of queues had been reached. In this case, you should consider the following alternatives:

  - Examine the SMALLQUEUEITEMS parameter to see whether the value has been significantly reduced from its default of 9999. This parameter can force queues to be converted to the large queue format if it has a large number of small items. It can be more efficient to write the items separately than to rewrite the whole small queue data area each time.

    If the SMALLQUEUEITEMS parameter has been significantly reduced, verify that the value is appropriate for the TS pool. If the value is appropriate, consider specifying a larger value for the TSLSTFUL guidance variable in USOURCE(CICGUIDE) as described below.

  - Examine the SMALLQUEUESIZE parameter to see whether the value has been significantly reduced from its default of 32K. This parameter can force queues to be converted to the large queue format at a smaller size than 32K. This action might be desired to prevent large amounts of data being written to the small queue format.

    If the SMALLQUEUESIZE parameter has been significantly reduced, verify that the value is appropriate for the TS pool. If the value is appropriate, consider specifying a larger value for the TSLSTFUL guidance variable in USOURCE(CICGUIDE) as described below.

- Change the TSLSTFUL guidance variable in USOURCE(CICGUIDE) so Rule CIC324 is produced only when you wish to be aware of a larger number of List Full situations. Unless you have a very unusual situation, this alternative should not be selected. You normally should be aware of List Full situations unless you are using non-default values for the SMALLQUEUEITEMS parameter or SMALLQUEUESIZE parameter to deliberately force a queue to be converted to the large queue format based on the number of elements in the queue or the maximum queue size.

- You can "turn off" the rule using the process described in Section 3 of this User Manual. However, this alternative is **not** recommended! You should always be aware of List Full situations.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a TS server region)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a TS server region)

Rule CIC325: Shared temporary data list structure was out of space

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that the shared temporary data list structure was out of space.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

If a task tries to write to temporary storage and there is no space available, message **DFHXQ0442** (*CF structure strname request failed, structure is full*) will be issued. The failing request is given a NOSPACE indication if it originated from a CICS API request. CICS normally suspends the task (although the task can regain control in this situation by using either a HANDLE CONDITION NOSPACE command, or the RESP or NOHANDLE option on the WRITEQ TS command). If suspended, the task normally is not resumed until some other task frees the necessary space in main storage or the VSAM data set.

If the task attempted to write to a shared temporary storage queue on the coupling facility and there was insufficient space to satisfy the request, the request to write to a shared temporary storage queue is retried every half second. This delay and retry is in anticipation that (1) automatic ALTER processing might make space available, or (2) other tasks release (delete) queue entries and make space available.

This delay can produce unexplained and elongated response delays, especially if the waiting task owns exclusive-use resources, in which case all other tasks needing those resources must also wait.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to determine whether a Structure Full condition occurred. CICXQ1 variable S1RSP7CT (List structure out of space) indicates that the list structure was full.

CPExpert produces Rule CIC325 when the number of Structure Full conditions is greater than the value specified by the **TSNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSNOSPCE** is 0, indicating that CPExpert should produce Rule CIC325 whenever any Structure Full conditions occurred.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Any queues which are no longer in use should be deleted so that the space can be reused.
- Review the *Automatic ALTER parameters* to determine whether these parameters provide sufficient control over the conditions under which the TS server attempts an automatic ALTER action when the structure becomes nearly full.
- If the structure is not at its maximum size, it may be possible to start an ALTER request to increase the size using the MVS SETXCF command.
- Review the POOLSIZE parameter to determine whether this parameter is less than the value specified for the structure in the Coupling Facility Resource Management (CFRM) policy. Increase the POOLSIZE parameter if appropriate.
- Review the STRUCTURE parameter and INITSIZE parameter in the CFRM policy to determine whether these parameters should be increased.
- Change the TSNOSPCE guidance variable in USOURCE(CICGUIDE) so Rule CIC325 is produced only when you wish to be aware of a larger number of Structure Full situations. This alternative is **not** recommended! You should always be aware of Structure Full situations.
- You can "turn off" the rule using the process described in Section 3 of this User Manual. However, this alternative is **not** recommended! You should always be aware of Structure Full situations.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a TS server region)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a TS server region)

Rule CIC326: A large percent of list data reads had to be repeated

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of list data reads had to be repeated because the data was larger than the default data transfer size.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The amount of impact depends on how many list data reads had to be repeated and the resulting overhead and delay.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

The default action with this program error is to terminate the task abnormally. However, the application can take remedial action if the HANDLE CONDITION command had been issued before the READQ TS command had been issued.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of list data reads which had to be repeated because the data was larger than the default data transfer size, using the following algorithm:

$$\textit{Percent list data rereads} = \frac{S1RRLCT}{S1RDLCT}$$

where S1RRLCT = Number of list data reads that had to be repeated
S1RDLCT = Number of list data reads

CPExpert produces Rule CIC326 when the percent of list data reads which had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTLDR** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLDR** is 0 indicating that CPExpert should produce Rule CIC326 when any list data reads must be repeated because the data was larger than the default data transfer size.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of shared temporary storage pools to determine whether the READQ TS commands can be issued with a correctly-sized data area for the INTO clause.

- Change the TSPCTLDR guidance variable in USOURCE(CICGUIDE) so Rule CIC326 is produced only when you wish to be aware of applications that cause a larger percent of list data reads must be repeated because the data was larger than the default data transfer size.

- You can specify **%LET TSPCTLDR = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent of list data reads which had to be repeated because the data was larger than the default data transfer size cannot be greater than 100), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: CICS/TS for OS/390 Release 1.1
CICS Application Programming Reference: Section 1.140 (READQ TS*)*
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.2
CICS Application Programming Reference: Section 1.140 (READQ TS*)*
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.3
CICS Application Programming Reference: Section 1.150 (READQ TS*)*
CICS System Programming Reference: Section 1.5 (Exception conditions)

CICS/TS for z/OS Release 2.1
CICS Application Programming Reference: CICS API Commands (READQ TS*)*
CICS System Programming Reference: Chapter 1 (Exception conditions)

CICS/TS for z/OS Release 2.2

CICS Application Programming Reference: CICS API Commands (READQ TS)

CICS System Programming Reference: Chapter 1 (Exception conditions)

Rule CIC327: A large percent of index data reads had to be repeated

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of index data reads had to be repeated because the data was larger than the default data transfer size.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The amount of impact depends on how many index data reads had to be repeated and the resulting overhead and delay.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

The default action with this program error is to terminate the task abnormally. However, the application can take remedial action if the HANDLE CONDITION command had been issued before the READQ TS command had been issued.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of index data reads which had to be repeated because the data was larger than the default data transfer size, using the following algorithm:

$$\textit{Percent index data rereads} = \frac{\textit{S1RRQCT}}{\textit{S1RDQCT}}$$

where	S1RRQCT	=	Number of queue index reads that had to be repeated
	S1RDQCT	=	Number of queue index reads

CPExpert produces Rule CIC327 when the percent of queue index reads had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **TSPCTIDR** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTIDR** is 0, indicating that CPExpert should produce Rule CIC327 when any queue index reads must be repeated because the data was larger than the default data transfer size.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of shared temporary storage pools to determine whether the READQ TS commands can be issued with a correctly-sized data area for the INTO clause.

- Change the TSPCTIDR guidance variable in USOURCE(CICGUIDE) so Rule CIC327 is produced only when you wish to be aware of applications that cause a larger percent of queue index reads which must be repeated because the data was larger than the default data transfer size.

- You can specify **%LET TSPCTIDR = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent queue index reads which must be repeated because the data was larger than the default data transfer size cannot be greater than 100), or you can "turn off" the rule using the process described in Section 3 of this User Manual.

Reference: CICS/TS for OS/390 Release 1.1
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.2
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.3
CICS Application Programming Reference: Section 1.150 (READQ TS)
CICS System Programming Reference: Section 1.5 (Exception conditions)

CICS/TS for z/OS Release 2.1
CICS Application Programming Reference: CICS API Commands (READQ TS)
CICS System Programming Reference: Chapter 1 (Exception conditions)

CICS/TS for z/OS Release 2.2

CICS Application Programming Reference: CICS API Commands (READQ TS)

CICS System Programming Reference: Chapter 1 (Exception conditions)

Rule CIC328: High percent version check failed for an entry being updated

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of version check failed for an entry being updated, indicating that another task had updated the entry.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The CICS temporary storage control facility provides the application programmer with the ability to store data in temporary storage queues, either in main storage, in auxiliary storage on a direct_access storage device, or in a temporary storage data sharing pool. With shared temporary storage, applications can:

- Write data to a temporary storage queue (WRITEQ TS command)
- Update data in a temporary storage queue (WRITEQ TS REWRITE command)
- Read data from a temporary storage queue (READQ TS command)
- Read the next data from a temporary storage queue (READQ TS NEXT command)
- Delete a temporary storage queue (DELETEQ TS command)

The VERSCOMPARE keyword of the IXLLSTE or IXLLSTM macro specifies whether version number comparison is to be performed to determine whether the list entry should be processed. Version number comparison compares the version number of the designed list entry to a specified version number value. Version number comparison can request that the version number of the designated list entry be equal or less than or equal to a specified version number value. If the criterion is not met, the request is terminated.

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke implicit enqueuing. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

After issuing the READQ TS command, if the application wishes to modify the information and then issue a WRITEQ TS command, it is possible that the temporary storage queue record would have been updated by another task. In this case, the WRITEQ TS command would fail because of a version check.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of requests that encountered a version check failed for an entry being updated condition, using the following algorithm:

$$\text{Percent version check failed} = \frac{S1RSP4CT}{S1RSP1CT + \text{Abnormal responses}}$$

where S1RSP1CT = Number of normal responses
Abnormal = S1RSP2CT + S1RSP3CT + S1RSP4CT + S1RSP5CT+ S1RSP6CT + S1RSP7CT + S1RSP8CT

S1RSP2CT = Requests timed out by CF
S1RSP3CT = Specified entry not found
S1RSP4CT = Version check failed for entry being updated
S1RSP5CT = List authority comparison failed
S1RSP6CT = Maximum list key reached
S1RSP7CT = List structure was out of space
S1RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC328 when the percent requests that encountered a “version check failed for an entry being updated” condition is greater than the value specified by the **TSPCTVCF** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTVCF** is 0.1 indicating that CPExpert should produce Rule CIC328 whenever more than one tenth percent of the requests encountered a “specified entry (queue or item) was not found” condition.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Modify the task to use explicit enqueuing on temporary storage queues where concurrently executing tasks can read and change queue(s) with the same temporary storage identifier. CICS provides the following explicit enqueuing commands:
  - EXEC CICS ENQ RESOURCE

- EXEC CICS DEQ RESOURCE

  These commands can be used to protect a temporary storage queue from being read and updated concurrently.

- Change the TSPCTVCF guidance variable in USOURCE(CICGUIDE) so Rule CIC328 is produced only when you wish to be aware of a larger percent of requests that experienced a version check failed for an entry being updated condition.

- You can specify **%LET TSPCTVCF = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent version check failed for an entry being updated cannot be greater than 100), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS Recovery and Restart Guide:
Section 3.4.5.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.5.5 Explicit enqueuing (by the application programmer)

CICS/TS for OS/390 Release 1.2
CICS Recovery and Restart Guide:
Section 3.4.4.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.4.5 Explicit enqueuing (by the application programmer)

CICS/TS for OS/390 Release 1.3
CICS Recovery and Restart Guide:
Section 3.4.4.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.4.5 Explicit enqueuing (by the application programmer)

CICS/TS for z/OS Release 2.1
CICS Recovery and Restart Guide: Chapter 13. Programming considerations
Implicit enqueuing on recoverable temporary storage queues
Explicit enqueuing (by the application programmer)

CICS/TS for z/OS Release 2.2
CICS Recovery and Restart Guide: Chapter 13. Programming considerations
Implicit enqueuing on recoverable temporary storage queues
Explicit enqueuing (by the application programmer)

Rule CIC330: High percent shared TS queue index buffers were in use

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a high percent of shared temporary storage (TS) queue index buffers were in use.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. It is provided as an "early warning" of a potential problem.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITEQ TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

The queue index buffer pool holds recently accessed index entries in storage to reduce significantly the coupling facility I/O for queue items. It is much more efficient to reuse an entry in the queue index buffer pool than to request the information for the coupling facility.

The queue index buffer pool is used for data associated with queue index entries if the total queue size does not exceed 32K bytes (that is, the TS queue is a "small queue").

When a request for the same queue arrives, the shared TS queue server determines whether the queue index information is in the buffer. If the information is in the buffer, a coupling facility access is avoided. When the request completes, the shared TS queue server places the information into a buffer, onto a least recently used (LRU) chain. If all other buffers are in use, a request for a new buffer will discard the contents of the least recently used buffer and reuse the storage as a free buffer.

The number of buffers in the queue index buffer pool is defined using the BUFFERS= keyword in the TS queue server parameters. The default specification is BUFFERS={100}, which specifies that 100 buffers should be allocated to the server address space. The maximum specification is 999999[1] buffers.

If a large percent of buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner.

Shared temporary storage queue server buffer pool statistics available in MXG file CICXQ2. CPExpert uses data in CICXQ2 to calculate the percent of queue index buffer pool buffers that were used, using the following algorithm:

$$\textit{Percent queue index buffer pool buffers used} = \frac{S2BFENTH}{S2BFQTY}$$

where S2BFENTH = Maximum number of queue index buffer pool buffers used
S2BFQTY = Number of buffers defined for the queue index buffer pool

CPExpert produces Rule CIC330 when the percent queue index buffer pool buffers that were used is greater than the value specified by the **TSPCTFBP** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTFBP** is 75, indicating that CPExpert should produce Rule CIC330 whenever more than 75% of the queue index buffer pool buffers were used.

This finding is produced as an "early warning" of a potential problem. If all buffers in the queue index buffer pool are used, the shared TS queue server begins to discard and reuse the oldest (least recently used) buffer. A large percent of LRU activity would be reported by **Rule CIC331**, and could imply "thrashing" in the queue index buffer pool.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the number of buffers specified for the queue index buffer pool. This action would particularly be appropriate if the finding shows an **increasing** percent of buffers used in the queue index buffer pool.

- Review applications using shared temporary storage to determine whether the applications are freeing temporary storage buffer pool buffers in a timely manner.

[1] IBM states that it is not worth defining extra buffers beyond the point where the definition might cause MVS paging, as it is more efficient to reread the index entry than to page in the buffer from auxiliary storage.

- Change the TSPCTFBP guidance variable in USOURCE(CICGUIDE) so Rule CIC330 is produced only when you wish to be aware of a larger percent of buffers used in the queue index buffer pool.

- You can specify **%LET TSPCTFBP = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent of buffers used in the queue index buffer pool cannot be greater than 100%), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: *CICS/TS Release 1.1*
CICS System Definition Guide: Section 3.4.3.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS Release 1.2
CICS System Definition Guide: Section 3.4.3.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS Release 1.3
CICS System Definition Guide: Section 4.2.2.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

Rule CIC331: High percent LRU activity in the TS queue index buffer pool

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that there was a high percent of requests that discarded and reused the oldest valid buffer using the Least Recently Used (LRU) algorithm in the shared temporary storage queue index buffer pool.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITEQ TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

The queue index buffer pool holds recently accessed index entries in storage to reduce significantly the coupling facility I/O for queue items. It is much more efficient to reuse an entry in the queue index buffer pool than to request the information for the coupling facility.

The queue index buffer pool is used for data associated with queue index entries if the total queue size does not exceed 32K bytes (that is, the TS queue is a “small queue”).

When a request for the same queue arrives, the shared TS queue server determines whether the queue index information is in the buffer. If the information is in the buffer, a coupling facility access is avoided. When the request completes, the shared TS queue server places the information into a buffer, onto a least recently used (LRU) chain. If all other buffers are in use, a request for a new buffer will discard the contents of the least recently used buffer and reuse the storage as a free buffer.

There is nothing intrinsically wrong with the shared TS queue server discarding the contents of the least recently used buffer. This is how the algorithms are designed. However, if a buffer is required from the LRU chain, this means that all other buffers were in use. If all other buffers were in use, it is possible that (1) an inadequate number of buffers were defined or (2) the application is not freeing buffers in a timely manner. In either case, a large percent of LRU activity could imply “thrashing” in the queue index buffer pool.

Shared temporary storage queue server buffer pool statistics available in MXG file CICXQ2. CPExpert uses data in CICXQ2 to calculate the percent LRU activity, using the following algorithm:

$$Percent\ LRU\ activity = \frac{S2BFGLRS}{S2BFGETS + S2BFPUTS}$$

where S2BFGLRS = Requests that discarded and reused oldest valid buffer
S2BFGETS = Requests to read a buffer
S2BFPUTS = Requests to write a buffer

CPExpert produces Rule CIC331 when the percent LRU activity is greater than the value specified by the **TSPCTLRU** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRU** is 0.1, indicating that CPExpert should produce Rule CIC331 whenever the percent LRU activity is greater than 0.1% of the queue index buffer pool requests. This low percent LRU activity was set as a default to alert you to a potential problem that requires attention.

Suggestion: If this finding is consistently produced, you should consider the following alternatives:

- Review the output from CIC331 over several days to determine whether the percent LRU activity is small and relatively stable. If so, you may wish change the TSPCTLRU guidance variable in USOURCE(CICGUIDE), so Rule CIC331 is produced only when you wish to make a change in the number of buffers. Increasing the guidance percent normally would be done only if you have discovered that the use of buffers is an expected result of application activity.

- Use the BUFFERS primary parameter for the queue server region, to increase the number of buffers specified for the queue index buffer pool. This action would particularly be appropriate if the finding shows an **increasing** number of requests that discarded and reused the oldest valid buffer.

- Review applications using shared temporary storage to determine whether the applications are freeing temporary storage buffer pool buffers in a timely manner.

- Change the TSPCTLRU guidance variable in USOURCE(CICGUIDE) so Rule CIC331 is produced only when you wish to be aware of a larger percent of LRU activity in the queue index buffer pool.

- You can specify **%LET TSPCTLRU = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent LRU activity cannot be greater than 100%), or you can "turn off" the rule using the process described in Section 3 of this User Manual. This alternative is NOT recommended, since you should normally be aware of potential "thrashing" in the queue index buffer pool.

Reference: *CICS/TS Release 1.1*
CICS System Definition Guide: Section 3.4.3.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS Release 1.2
CICS System Definition Guide: Section 3.4.3.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS Release 1.3
CICS System Definition Guide: Section 4.2.2.4: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21: Primary parameters
CICS Performance Guide: Shared TS queue server: buffer pool statistics

Rule CIC332: Excess buffers defined for TS queue index buffer pool

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that excess buffers had been defined for the shared temporary storage queue index buffer pool.

Impact: This finding has a LOW IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: A queue index buffer holds a queue index entry plus up to 32K of queue data (for a small queue). When a READ or WRITE request completes, the queue index information is retained in the buffer. Retaining the queue index information in the buffer can avoid the need to reread the queue index if the same queue is referenced from the same MVS image before the buffer has been reused.

The shared temporary storage queue server uses the queue index buffer pool to read and write queue index entries. The queue index buffer pool also is used for data associated with queue index entries if the total queue size does not exceed 32K bytes. The queue index buffer pool holds recently accessed index entries in storage to reduce coupling facility I/O.

The number of buffers in the queue index buffer pool is defined using the BUFFERS= keyword in the TS queue server parameters. The default specification is BUFFERS={100}, which specifies that 100 buffers should be allocated to for the server address space. The maximum specification is 999999, but IBM states that it is not worth defining extra buffers beyond the point where the definition might cause MVS paging, as it is more efficient to reread the index entry than to page in the buffer from auxiliary storage.

CPExpert analyzes the number of buffers used in the queue index buffer pool to determine whether excess buffers were defined. CPExpert concludes that excess buffers were defined if a small percent of the buffers in the queue index buffer pool were used.

Shared temporary storage queue server buffer pool statistics are available in MXG file CICXQ2. CPExpert uses data in CICXQ2 to calculate the maximum percent of the queue index buffer pool buffers that had been used, using the following algorithm:

$$\textit{Maximum percent queue index buffer pool buffers used} = \frac{S2BFENTH}{S2BFQTY}$$

where S2BFENTH = Maximum number of queue index buffer pool buffers used
S2BFQTY = Number of buffers defined for the queue index buffer pool

CPExpert produces Rule CIC332 when the percent queue index buffer pool buffers used is less than the value specified by the **TSPCTUSE** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTUSE** is 10, indicating that CPExpert should produce Rule CIC332 whenever less than 10% of the queue index buffer pool buffers were used. CPExpert suppresses this finding if 100 buffers or less are defined.

Suggestion: If this finding is produced **consistently**, you should consider the following alternatives:

- Reduce the number of buffers specified for the queue index buffer pool.
- Change the TSPCTUSE guidance variable in USOURCE(CICGUIDE) so Rule CIC332 is produced only when you wish to make a change in the number of buffers.
- You can specify **%LET TSPCTUSE = 0;** in USOURCE(CICGUIDE) to suppress this finding (the percent used cannot be less than zero), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3.4 (Primary parameters)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3.4 (Primary parameters)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2.4 (Primary parameters)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC333: High percent waits on shared temporary storage buffer pool lock

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that there was a high percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Shared temporary storage queues are stored in named pools in an MVS coupling facility. A shared TS pool consists of an XES list structure, which is accessed through a cross-memory queue server region.

There are two situations in which shared temporary storage requests can wait on the shared TS pool:

- **The queue pool is locked for exclusive use**. Message **DFHXQ0407** (CF structure strname is not available for shared use) issued if a request for a TS pool cannot be satisfied because the queue pool is locked for exclusive use by some other job (such as a queue pool unload or reload job).

  The server is terminated in this situation. Consequently, this situation is unlikely to be the cause of frequent waits on a shared temporary storage buffer pool lock.

- **SUSPEND on resource type TSPOOL**. Resource type TSPOOL indicates that the maximum number of concurrent requests (10) for a temporary storage pool in the coupling facility has been reached. The task resumes when one of the requests completes.

Shared temporary storage queue server buffer pool statistics available in MXG file CICXQ2. CPExpert uses data in CICXQ2 to calculate the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock, using the following algorithm:

$$\textit{Percent TS requests that waited on a buffer pool lock} = \frac{\textit{S2BFPWTS}}{\textit{Total TS requests}}$$

where S2BFPWTS = Maximum number of queue index buffer pool buffers used
TS requests = S2BFGETS + S2BFPUTS + S2BFFRES

S2BFGETS = Requests to get a buffer
S2BFPUTS = Requests to put back buffer with valid contents
S2BFFRES = Requests to put back a buffer as empty

CPExpert produces Rule CIC333 when the percent of requests to the shared temporary storage buffer pool that waited on a buffer pool lock is greater than the value specified by the **TSPCTWBP** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWBP** is 0.1, indicating that CPExpert should produce Rule CIC333 whenever more than one tenth of the requests waited on a buffer pool lock.

Suggestion: If this finding is produced often, you should consider the following alternatives:

- Verify that tasks that are performing operations on the same temporary storage queue are intended to do so. Possibly, the ID of the queue is unintentionally not unique.

- If possible, create more temporary storage queues to reduce the contention between tasks.

- If a task has made an unusually large number of shared temporary storage requests, it could be looping. You should verify that the tasks are not looping.

- Change the TSPCTWBP guidance variable in USOURCE(CICGUIDE) so Rule CIC333 is produced only when you wish to be aware of a larger number of situations when requests wait on a buffer pool lock.

- You can "turn off" the rule using the process described in Section 3 of this User Manual. However, this alternative is **not** recommended! You should always be aware of situations when requests wait on a buffer pool lock.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS Problem Determination Guide: Section 2.3.12.6 (Resource type TSPOOL)

CICS/TS for OS/390 Release 1.2
CICS Problem Determination Guide: Section 2.3.12.6 (Resource type TSPOOL)

CICS/TS for OS/390 Release 1.3
CICS Problem Determination Guide: Section 2.3.12.6 (Resource type TSPOOL)

CICS/TS for z/OS Release 2.1

CICS Problem Determination Guide: Chapter 6 (Dealing with waits: Resource type TSPOOL)

CICS/TS for z/OS Release 2.2

CICS Problem Determination Guide: Chapter 6 (Dealing with waits: Resource type TSPOOL)

Rule CIC334: High percent GET waits on shared temporary storage buffer lock

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that there was a high percent of GET requests to the shared temporary storage buffer pool that waited on a buffer lock.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The shared temporary storage queue server uses a *queue index buffer pool* within its region, to read and write queue index entries. When a READQ TS or WRITEQ TS request completes, the queue index information is retained in the buffer. Retaining the queue index entries in the queue index buffer pool can avoid the need to reread the queue index entry if the same queue is referenced from the same MVS image before the buffer has been reused.

Temporary storage control commands WRITEQ TS and DELETEQ TS invoke *implicit enqueuing*. However, CICS enqueuing is not invoked for READQ TS commands. This makes possible for one task to read a temporary storage queue record while another is updating the same record.

CICS provides two explicit enqueuing commands (EXEC CICS ENQ RESOURCE and EXEC CICS DEQ RESOURCE). These commands can be used to protect a temporary storage queue from being read and updated concurrently.

After a task has issued an ENQ RESOURCE(data-area) command, any other task that issues an ENQ RESOURCE command with the same data-area parameter is suspended until the task issues a matching DEQ RESOURCE(data-area) command, or until the UOW ends.

Shared temporary storage queue server buffer pool statistics are available in MXG file CICXQ2. Variable S2BFLWTS report the number of GET waits for a buffer because another request owned the buffer. CPExpert uses data in CICXQ2 to calculate the percent of GET waits on buffer lock, using the following algorithm:

$$\textit{Percent GET waits on buffer lock} = \frac{\textit{S2BFLWTS}}{\textit{S2BFGETS}}$$

where S2BFLWTS = Number of GET waits on buffer lock
S2BFGETS = Number of GET requests

CPExpert produces Rule CIC334 when the percent GET waits on buffer lock is greater than the value specified by the **TSPCTWBL** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTWBL** is 1, indicating that CPExpert should produce Rule CIC334 whenever more than one percent of the GET requests waited for a buffer lock. CPExpert suppresses this finding unless there are more than 1,000 GET requests in the statistics interval.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Examine the tasks using the shared temporary storage queue to see whether it possible to make tasks relinquish control of the buffer item more quickly. Consider reducing the size of UOWs, or making conversational tasks pseudoconversational.

- Change the TSPCTWBL guidance variable in USOURCE(CICGUIDE) so Rule CIC328 is produced only when you wish to be aware of a larger percent of GET requests that waited for buffer lock.

- You can specify **%LET TSPCTWBL = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent GET requests that wait for buffer lock cannot be greater than 100), or you can "turn off" the rule using the process described in Section 3 of this User Manual.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS Recovery and Restart Guide:
Section 3.4.5.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.5.5 Explicit enqueuing (by the application programmer)

CICS/TS for OS/390 Release 1.2
CICS Recovery and Restart Guide:
Section 3.4.4.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.4.5 Explicit enqueuing (by the application programmer)

CICS/TS for OS/390 Release 1.3
CICS Recovery and Restart Guide:
Section 3.4.4.3 Implicit enqueuing on recoverable temporary storage queues
Section 3.4.4.5 Explicit enqueuing (by the application programmer)

CICS/TS for z/OS Release 2.1

CICS Recovery and Restart Guide: Chapter 13. Programming considerations

Implicit enqueuing on recoverable temporary storage queues

Explicit enqueuing (by the application programmer)

CICS/TS for z/OS Release 2.2

CICS Recovery and Restart Guide: Chapter 13. Programming considerations

Implicit enqueuing on recoverable temporary storage queues

Explicit enqueuing (by the application programmer)

Rule CIC340: LOC=ANY request initially failed and was retried

Finding: The CICS Shared Temporary Storage (TS) Queue Server statistics showed that a high percent of LOC=ANY storage requests initially failed to obtain the requested storage, and requests were retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a specific named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for <u>each pool</u> defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the “compress attempt” to reduce fragmentation in the AXMPGANY storage pool will correspondingly increase.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried, using the following algorithm:

$$\textit{Percent AXMPGANY storage requests retried} = \frac{\textit{S3ANYRQC}}{\textit{S3ANYRQG}}$$

where S3ANYRQC = Times a storage request initially failed and was retried
S3ANYRQG = Number of storage GET requests

CPExpert produces Rule CIC340 when the percent of storage requests in the AXMPGANY pool initially failed and were retried is greater than the value specified by the **TSPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTARC** is .1, indicating that CPExpert should produce Rule CIC340 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

Suggestion: If this finding is produced, you should consider the following alternatives:

- A possible cause of this situation is that the specific shared temporary storage queue is used for varying sizes of data items (if the shared TS queue contained data items with identical sizes, storage in the AXMPGANY pool would not be fragmented). This situation could be caused by (1) a specific task using differing sizes for shared temporary storage items, or (2) different tasks using the shared TS queue and the different tasks have differing sizes of the shared temporary storage items.

  Consequently, one way to reduce the overhead caused by "compressing" the AXMPGANY pool is to review the tasks using the shared TS queue. Determine whether multiple shared TS queues can be used (with each queue having a similar size).

- If this finding occurs often (or if a large percent of requests initially failed and were retried), this could be an indication that storage in the AXMPGANY pool is becoming exhausted and requests might begin failing. Requests are retried only (1) if there are no free areas of the right size and (2) there is not enough storage left in the pool to satisfy the request. Either of these situations occurring frequently could indicate that storage in the pool is in danger of becoming exhausted. If storage is in danger of becoming exhausted, Rule CIC342 (LOC=ANY had low percent minimum free storage) might be produced, but you might have altered the guidance for Rule CIC342 and Rule CIC342 might have been suppressed. If storage in the AXMPGANY pool is becoming exhausted, you should consider the following alternatives:

  - Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

    The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for other storage areas.

  - Decrease the BUFFERS initialization parameter to leave more storage for request processing.

- You probably should take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

- You should normally NOT take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Change the TSPCTARC guidance variable in USOURCE(CICGUIDE) so Rule CIC340 is produced only when you wish to be aware of a different percent of storage requests in the AXMPGANY pool that initially failed and were retried. Since Rule CIC340 provides an "early warning" of potential impending request failures, you should not normally change the TSPCTARC guidance variable.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC341: LOC=ANY requests were unable to obtain storage and failed

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a high percent of LOC=ANY storage requests initially failed to obtain the requested storage, were retried after merging adjacent small free areas to form larger areas, and failed because the requested amount of storage was unavailable.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for <u>each pool</u> defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert produces Rule CIC341 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S3ANYRQS[1]) is greater than the value specified by the **TSANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **TSANYRQS** is 0, indicating that CPExpert should produce Rule CIC341 when any storage request in the AXMPGANY pool failed after retry.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for

[1] Please note that the *CICS Performance Guides* describe S3ANYRQF as failed requests and S3ANYRQS as requests to release storage. These descriptions were "reversed" initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S3ANYRQF is the requests to "free" or release storage and S3ANYRQS is number of failed requests because of "short on storage".

other storage areas. You might wish to increase the REGION parameter to larger values than the IBM recommendation if storage requests fail.

- Decrease the BUFFERS initialization parameter to leave more storage for request processing.

  - You probably **should** take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

  - You normally should **not** take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Examine the tasks using the shared temporary pool to determine whether some tasks could use a different shared temporary storage pool.

- Change the TSANYRQS guidance variable in USOURCE(CICGUIDE) so Rule CIC341 is produced only when you wish to be aware of a different number of storage requests in the AXMPGANY pool that failed after retry. Since Rule CIC341 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the TSANYRQS guidance variable**.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC342: LOC=ANY had low percent minimum free storage

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that the minimum amount of free storage for the CICS shared temporary storage server pool was low.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for <u>each pool</u> defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as the *AXMPGANY pool*. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as the *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGANY pool, using the following algorithm:

$$\textit{Minimum percent free AXMPGANY storage} = \frac{\textit{S3ANYLO}}{\textit{S3ANYMX}}$$

where S3ANYLO = Lowest amount of storage that has been free since reset
S3ANYSZ = Total pages in the storage pool

CPExpert produces Rule CIC342 when the percent free storage in the AXMPGANY pool is less than the value specified by the **TSPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTAMN** is 25, indicating that CPExpert should produce Rule CIC342 whenever less than 25% of storage in the AXMPGANY pool is free.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for other storage areas.

- Decrease the BUFFERS initialization parameter to leave more storage for request processing.

  - You probably should take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

  - You should normally NOT take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Change the TSPCTAMN guidance variable in USOURCE(CICGUIDE) so Rule CIC342 is produced only when you wish to be aware of a different minimum percent of free storage in the AXMPGANY pool.

.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC343: LOC=BELOW request initially failed and was retried

Finding: The CICS Shared Temporary Storage (TS) Queue Server statistics showed that a high percent of LOC=BELOW storage requests initially failed to obtain the requested storage, and requests were retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for <u>each pool</u> defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGLOW storage pool will correspondingly increase.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried, using the following algorithm:

$$\textit{Percent AXMPGLOW storage requests retried} = \frac{\textit{S3LOWRQC}}{\textit{S3LOWRQG}}$$

where S3LOWRQC = Times a storage request initially failed and was retried
S3LOWRQG = Number of storage GET requests

CPExpert produces Rule CIC343 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **TSPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRC** is .1, indicating that CPExpert should produce Rule CIC343 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

Suggestion: If this finding is produced, you should consider the following alternatives:

- A possible cause of this situation is that the specific shared temporary storage queue is used for varying sizes of data items (if the shared TS queue contained data items with identical sizes, storage in the AXMPGLOW pool would not be fragmented). This situation could be caused by (1) a specific task using differing sizes for shared temporary storage items, or (2) different tasks using the shared TS queue and the different tasks have differing sizes of the shared temporary storage items.

  Consequently, one way to reduce the overhead caused by "compressing" the AXMPGLOW pool is to review the tasks using the shared TS queue. Determine whether multiple shared TS queues can be used (with each queue having a similar size).

- If this finding occurs often (or if a large percent of requests initially failed and were retried), this could be an indication that storage in the AXMPGLOW pool is becoming exhausted and requests might begin failing. Requests are retried only (1) if there are no free areas of the right size and (2) there is not enough storage left in the pool to satisfy the request. Either of these situations occurring frequently could indicate that storage in the pool is in danger of becoming exhausted. If storage is in danger of becoming exhausted, Rule CIC345 (LOC=BELOW had low percent minimum free storage) might be produced, but you might have altered the guidance for Rule CIC345 and Rule CIC345 might have been suppressed. If storage in the AXMPGLOW pool is becoming exhausted, you should consider the following alternatives:

  - Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

    The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for other storage areas.

  - Decrease the BUFFERS initialization parameter to leave more storage for request processing.

  - You probably should take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

  - You should normally NOT take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Change the TSPCTLRC guidance variable in USOURCE(CICGUIDE) so Rule CIC343 is produced only when you wish to be aware of a different percent of storage requests in the AXMPGLOW pool that initially failed and were retried. Since Rule CIC343 provides an ”early warning” of potential impending request failures, you normally should not change the TSPCTLRC guidance variable.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC344: LOC=BELOW requests were unable to obtain storage and failed

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that there was a high percent of LOC=BELOW storage requests that initially failed to obtain the requested storage, were retried after merging adjacent small free areas to form larger areas, and failed because the requested amount of storage was unavailable.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for each pool defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert produces Rule CIC344 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S3LOWRQS[1]) is greater than the value specified by the **TSLOWRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **TSLOWRQS** is 0, indicating that CPExpert should produce Rule CIC344 when any storage request in the AXMPGLOW pool failed after retry.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for

[1]Please note that the *CICS Performance Guides* describe S3LOWRQF as failed requests and S3LOWRQS as requests to release storage. These descriptions are "reversed" and IBM Hursley has agreed to correct the *CICS Performance Guides*. S3LOWRQF is the requests to "free" or release storage and S3LOWRQS is number of failed requests because of "short on storage".

other storage areas. You might wish to increase the REGION parameter to larger values than the IBM recommendation if storage requests fail.

- Decrease the BUFFERS initialization parameter to leave more storage for request processing.

  - You probably **should** take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

  - You normally should **not** take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Examine the tasks using the shared temporary pool to determine whether some tasks could use a different shared temporary storage pool.

- Change the TSLOWRQS guidance variable in USOURCE(CICGUIDE) so Rule CIC344 is produced only when you wish to be aware of a different number of storage requests in the AXMPGLOW pool that failed after retry. Since Rule CIC344 describes a situation that has a high impact on the performance of the CICS region, **you normally should not change the TSLOWRQS guidance variable**.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC345: LOC=BELOW had low percent minimum free storage

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that there was a high percent of LOC=BELOW storage requests that initially failed to obtain the requested storage, but was retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for each pool defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as the *AXMPGANY pool*. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as the *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert uses data in CICXQ3 to calculate the minimum percent of free storage in the AXMPGLOW pool, using the following algorithm:

$$\textit{Minimum percent free AXMPGLOW storage} = \frac{S3LOWLO}{S3LOWMX}$$

where S3LOWLO = Lowest amount of storage that has been free since reset
S3LOWMX = Total number of pages in the storage pool

CPExpert produces Rule CIC345 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **TSPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLMN** is 25, indicating that CPExpert should produce Rule CIC345 whenever less than 25% of storage in the AXMPGLOW pool is free.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for other storage areas.

- Decrease the BUFFERS initialization parameter to leave more storage for request processing.

  - You probably should take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

  - You should normally NOT take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Change the TSPCTLMN guidance variable in USOURCE(CICGUIDE) so Rule CIC345 is produced only when you wish to be aware of a different minimum percent of free storage in the AXMPGLOW pool.

.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

Rule CIC401: Adds were rejected because shared data table was full

Finding: The CICS interval statistics showed that adds to a shared data table were rejected because the shared data table was full.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: For VSAM data sets, CICS file control provides applications with *file control commands* that read, update, add, and browse the data sets. There are many features provided by IBM that allow applications to use file control commands to reference VSAM data sets, and allow separate applications to *share* the data. One such feature is the *shared data table* approach.

An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

With shared data table support, the KSDS file is called the *source data set*. The copy of the file in memory is called the *data table*. The process of copying the records from the file to the data table is called *loading the data table*. Whenever a CICS application wishes to reference the VSAM file using normal file control commands, CICS attempts to use the representation of the file in the data table, rather than accessing the source data.

If applications are running in different CICS regions on the MVS image, the region that initially opens a file assigned to a shared data table “owns” the data table. The CICS region that owns the data table is known as a *file-owning region (FOR),* and is responsible for file access to both the data table and (if necessary) to the VSAM file.

Any other CICS region that uses the file is known as an *application-owning region (AOR)*. The AOR does not need to take any special action to reference the data table; the linking of the file name used by the AOR to the data table in the FOR is established when the file is defined[1].

When the first file that is defined as a data table is opened in an FOR, the FOR attempts to register itself as a shared data table (SDT) server. This operation is performed automatically and is known as an SDT LOGON. The opening of the file can be caused by the FOR or by the AOR that first accesses the file.

In the FOR, the file is known as a *local file* and, in the AOR, the file is known as a *remote file*. The same region can be both an FOR with some data tables and an AOR for others (that is, a region can own a data table in its own region which it and other regions can access, and the region can also use data tables in other FORs).

CICS supports two types of data table :

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with its source data set. Any update or delete action on a record in the CICS-maintained data table is automatically applied to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set. After loading of the data table has completed, all file control commands that access the filename are performed only on the data table.

Records are placed into a shared data table in one of three ways: (1) records are placed in the data table by the initial loading of the data table from the source data set, (2) records are subsequently added to the data table from the source data set, and (3) new records are written to the data table after the data table has been loaded.

- **Initial loading of the data table**. During initial loading of the data table, CICS reads the entire VSAM KSDS file and attempts to place all records in the data table. The XDTRD user exit can be used to limit the records that are placed in the data table (using screening criteria appropriate to the applications sharing the data). If all records that pass the screening

[1] A data table is defined by means of the CEDA DEFINE FILE command.

criteria (if any) will not fit into the data table[2], a “table full” condition applies during the initial loading of the data table.

- **Records subsequently added from the source data set**. Records from a VSAM KSDS file can be added to a data table after the initial loading[3] if either (1) records would not fit into the data table because the table became full during initial loading or (2) the user exit had excluded records during initial loading but an excluded record was subsequently required by an application. If the data table had become full during initial loading, some record must be deleted from the data table for another record to be subsequently added from the source data set.

- **New records written to the data table.** New records can be added to a data table via the WRITE file control command. The XDTAD global user exit program can be used to limit the records that are placed in the CFDT as a result of a WRITE request issued to a data table (using screening criteria appropriate to the applications sharing the data). If the data table is at its maximum number of records when a WRITE is attempted, a “table full” condition applies[4].

Regardless of how records are added to a shared data table, the number of records in the table cannot exceed the number specified by the MAXNUMRECS variable specified for the table. If that limit is reached during loading or while adding to the table, a “table full” condition applies. Encountering a “table full” condition can have several undesirable effects.

- The AOR requesting the record must function ship the request to the FOR, and the record must be retrieved from the source data set[5]. Retrieving the record from the source data set requires more overhead and more elapsed time than retrieval from the data table. Additionally, function shipping requires significantly more overhead than does cross-memory services.

  On the FOR, the source data set would be accessed to add the records to the data table. However, since the data table was at its maximum number of records, the records would then be rejected (unless some record had been deleted from the data table while the function-ship

[2] The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be placed in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that the entire VSAM KSDS file can be placed in the data table (subject, of course, to screening criteria applied by the XDTRD user exit).

[3] Records from the source data set cannot be added to a user-maintained data table after the initial loading of the data table. User-maintained data tables are separated from the source data set after loading.

[4] Note that this situation would not normally occur if the record had been updated, since a “read for update” would have been issued to obtain a lock on the record, and the record would simply be re-written to the table.

[5] This description applies only to CICS-maintained data tables. As mentioned earlier, a user-maintained data table is not associated with its source data set after the table is loaded.

process was executed). This sequence causes considerable overhead, and would result in performance degradation.

- If the data table size specified in the file definition is significantly less than the number of records in the source data set, only a small part of the file is loaded. Potentially, many function-shopped requests could be sent by the AOR to the FOR, not only creating unnecessary overhead but also delaying access to the records and causing increased response.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTATF variable to assess the number of records that CICS attempted to add to the table but was unable to do so because the table was full. This count means that the data table already contained the maximum number of records specified in the MAXNUMRECS parameter of the DEFINE FILE command.

CPExpert produces Rule CIC401 when the A17DTATF value is greater than the **SDTFULL** guidance variable in USOURCE(CICGUIDE). The default value for the **SDTFULL** guidance variable is one, indicating that CPExpert should produce Rule CIC401 when more than one shared data table full condition was detected. Note that the initial loading of the shared data table normally will produce a count of one for the A17DTATF, so the default value of the SDTFULL variable accounts for this situation.

Many of the above comments apply to a *Coupling Facility Data Table* (CFDT). If CFDT information exists (in MXG CICCFS6D, CICCFS6D, CICCFS8D, and CICCFS9D), CPExpert will analyze CFDT performance problems. Since a CFDT is a data table (residing in a coupling facility rather than an MVS data space), the data table statistics in CICFCR apply to a CFDT as well as a normal shared data table.

Rule CIC400(series) report data table information regardless of whether the data table resides in an MVS data space or in a coupling facility. Rule CIC420(series) report data table information when the data table resides in a coupling facility.

CPExpert analyzes the CFDT coupling facility statistics with respect to a CFDT exceeding the MAXNUMRECS value. Rule CIC424 (*MAXNUMRECS was reached for CFDT*) will be produced to report information on a coupling facility basis, if the number of records in a CFDT exceeds the MAXNUMRECS value specified for the CFDT. Additionally, CPExpert will produce Rule CIC401 to identify the data table in the coupling facility, and will report information about the CFDT just as though it were a data table in an MVS data space. This point is particularly important since the CFDT statistics to not identify a specific CFDT that exceeds the MAXNUMRECS value.

Suggestion: If Rule CIC401 is produced regularly (and particularly if the shared data table full count is high), you consider the following alternatives:

- Increase the number of records allowed in the shared data table. The number of records allowed in the shared data table is controlled by MAXNUMRECS parameter of the DEFINE FILE command. The DEFINE FILE command can be applied at the CEDA panel.

  Increasing the number of records in the shared data table would result in a larger table, using more virtual storage. Using more virtual storage could increase CICS paging operations, or increase overall system paging. Consequently, paging rates should be monitored if there is a large increase in the number of records allowed in the shared data table.

- If increasing the number of records allowed in the shared data table is not feasible, determine whether the XDTRD user exit has been used for the VSAM KSDS file. The **EXIT** column of the Rule CIC401 output will show the number of records rejected by the XDTRD or XDTAD user exit.

  If the XDTRD user exit has **not** been used during initial loading, consider using the XDTRD user exit and devising appropriate screening criteria to be applied to the VSAM KSDS records as they are read by CICS during initial loading of the table. If the XDTRD user exit **has** been used, consider revising the screening criteria in the user exit to exclude a larger number of records during loading.

- If increasing the number of records allowed in the shared data table is not feasible, determine whether (1) the shared data table had a significant number of WRITE requests and (2) whether the XDTAD user exit has been used. If there was a significant number of WRITE requests and the XDTAD user exit has **not** been used, consider adding the XDTAD user exit and devising appropriate screening criteria to be applied to the VSAM KSDS records as they are written to the data table. If the XDTAD user exit **has** been used, consider revising the screening criteria in the user exit to exclude a larger number of records during loading.

  Both the above options should be evaluated with the goal of reducing the number of records that are loaded into the shared data table so some space is available for added records.

- Examine the applications accessing the shared data table to determine whether records can be deleted (using the DELETE command) from the table, or whether they can be deleted earlier. This command can be used with CICS-maintained shared data tables only[6] if *the record in the VSAM file should be deleted*.

- You can change the **SDTFULL** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC401 is produced too often. If you chose this option, please send a note to Don_Deese@cpexpert.com to explain your reason, so I can better appreciate why a site would knowingly experience table full conditions.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

[6] With CICS-maintained shared data tables, file update actions against the shared data table will be applied to the file before they are applied to the table.

Rule CIC402: Records not found in CICS-maintained shared data table

Finding: The CICS interval statistics showed that a large number of records were not found in a CICS-maintained shared data table.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

With shared data table support, the KSDS file is called the *source data set*. The copy of the file in memory is called the *data table*. The process of copying the records from the file to the data table is called *loading the data table*. Whenever a CICS application wishes to reference the VSAM file using normal file control commands, CICS attempts to use the representation of the file in the data table, rather than accessing the source data.

CICS supports two types of data table:

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is automatically applied to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Records are placed into a data table in one of three ways: (1) records are placed in the data table by the initial loading of the data table from the source data set, (2) records are subsequently added to the data table from the source data set, and (3) new records are written to the data table after the data table has been loaded..

- **Initial loading of the data table**. During initial loading of the data table, CICS reads the entire VSAM KSDS file and attempts to place all records in the data table. The XDTRD user exit can be used to limit the records that are placed in the data table (using screening criteria appropriate to the applications sharing the data). If all records that pass the screening criteria (if any) will not fit into the data table[1], a “table full” condition applies during the initial loading of the data table.

- **Records subsequently added from the source data set**. Records from a VSAM KSDS data set can be added to a data table after the initial loading[2] if either (1) records would not fit into the data table because the table became full during initial loading or (2) the XDTRD user exit had excluded records during initial loading but an excluded record was subsequently required by an application. If the data table had become full during initial loading, some record must be deleted from the data table for another record to be subsequently added from the source data set.

- **New records written to the data table.** New records can be added to a data table via the WRITE file control command. The XDTAD global user exit program can be used to limit the records that are placed in the CFDT as a result of a WRITE request issued to a data table (using screening criteria appropriate to the applications sharing the data). If the data table is at its maximum number of records when a WRITE is attempted, a “table full” condition applies[3].

Applications can reference the data table during initial loading. These references will produce a “record not found” condition if the references are to records outside the range of those already loaded into the data table. For both CICS-maintained data tables and user-maintained data tables, the “record not found” condition can occur during initial loading. These conditions should be small and can normally be ignored.

For user-maintained data tables[4], the “record not found” condition should not occur after loading of the data table.

For CICS-maintained data tables, small numbers of the “record not found”

[1] The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be placed in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that the entire VSAM KSDS file can be placed in the data table (subject, of course, to screening criteria applied by the XDTRD user exit).

[2] Records from the source data set cannot be added to a user-maintained data table after the initial loading of the data table. User-maintained data tables are separated from the source data set after loading.

[3] Note that this situation would not normally occur if the record had been updated, since a “read for update” would have been issued to obtain a lock on the record, and the record would simply be re-written to the table.

[4] If the “record not found” condition does occur after initial loading of a user-maintained data table, there probably is a logic or coding error with the application. These situations are analyzed in Rule CIC403.

condition after initial loading should not normally be a cause for alarm. However, a large number of “records not found” will cause unnecessary overhead and delay to applications.

- The AOR requesting the record must function ship the request to the FOR, and the record must be retrieved from the source data set. Retrieving the record from the source data set requires more overhead and more elapsed time than retrieval from the data table. Additionally, function shipping requires significantly more overhead than does cross-memory services.

  On the FOR, the source data set would be accessed to retrieve the record. Accessing the source data set for a large number of records indicates that the data table is not operating efficiently.

- If the data table size specified in the file definition is significantly less than the number of records in the source data set, only a small part of the file is loaded. Potentially, many function-shopped requests could be sent by the AOR to the FOR, not only creating unnecessary overhead but also delaying access to the records and causing increased response.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of times records were not found for CICS-maintained data tables. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data table; CICS was required to retrieve the record from the source data set.

The CICS statistics contain an indicator (the A17DTTYP variable) describing whether a shared data table is CICS-maintained or user-maintained. Unfortunately, this indicator is set only when the VSAM data set is closed. CPExpert does a “reverse scan” of the shared data table statistics to identify CLOSED status of a shared data table. A17DTTYP is propagated to all observations prior to CLOSE, so CPExpert can identify whether a shared data table is CICS-maintained or User-maintained. Additionally, CPExpert examines all variables that represent data set accesses that modify the data set (these are the A17DSGU, A17DSWRU, A17DSWRA, A17DSDEL, A17RMDEL, and A17DSBRU variables). If the source data set is modified, the data table must be a CICS-maintained dat table.

This approach works only if the shared data table actually was closed within the CICS statistics data available in the performance data base being analyzed by CPExpert, or if modifications were made to the data set. If the shared data table CLOSE record is not present in the data or if modifications were made to the data set, CPExpert will be unable to determine whether the shared data table is CICS-maintained or user-maintained. From a practical matter, this should be of little import as the analysis depending on CICS-maintained versus user-maintained data tables would not be seriously effected for tables that are not closed and which have no source data set access.

CPExpert produces Rule CIC402 when the A17DTRNF value is greater than the **CICSRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CICSRNF** guidance variable is 100, indicating that CPExpert should produce Rule CIC402 when more than one hundred read requests resulted in a "record not found" condition. Note that during initial loading of the data table, any read requests outside the range of records loaded will produce a count for the "record not found" condition. As mentioned earlier, this is not normally a cause for alarm.

Suggestion: A large number of "records not found" can indicate that (1) the data table is too small for the VSAM file, or (2) the screening criteria in the XDTRD user exit does not match the record-referencing pattern of applications. If Rule CIC402 is produced regularly, you consider the following alternatives:

- Increase the number of records allowed in the shared data table. The number of records allowed in the shared data table is controlled by MAXNUMRECS parameter of the DEFINE FILE command. The DEFINE FILE command can be applied at the CEDA panel.

  Increasing the number of records in the shared data table would result in a larger table, using more virtual storage. Using more virtual storage could increase CICS paging operations, or increase overall system paging. Consequently, paging rates should be monitored if there is a large increase in the number of records allowed in the shared data table.

- If increasing the number of records allowed in the shared data table is not feasible, determine whether the XDTRD user exit has been used for the VSAM KSDS file. If the XDTRD user exit **has** been used, consider revising the screening criteria in the user exit to include a larger number of records during loading. The screening criteria might not match the records required by the application (perhaps because of different transactions or different distribution of transaction types requiring different records). Consequently, the existing screening criteria might exclude more records than are required by the applications or transaction patterns being submitted.

- Alternatively, the application might "test" for the presence of a record before adding a new record. If this is the case, the CICSRNF guidance should be changed for this specific table (Section 3 describes how to change guidance for individual shared data tables).

- You can change the CICSRNF guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC402 is produced too often.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

Rule CIC403: Records not found in user-maintained shared data table

Finding: The CICS interval statistics showed that a high percent of records were not found in a shared data table.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

With shared data table support, the KSDS file is called the *source data set*. The copy of the file in memory is called the *data table*. The process of copying the records from the file to the data table is called *loading the data table*. Whenever a CICS application wishes to reference the VSAM file using normal file control commands, CICS attempts to use the representation of the file in the data table, rather than accessing the source data.

CICS supports two types of data table :

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is *automatically applied* to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

Records are placed into a data table in one of three ways: (1) records placed in the data table by the initial loading of the data table from the source data set, (2) records subsequently added to the data table from the source data set, and (3) new records written to the data table after the data table has been loaded..

- **Initial loading of the data table**. During initial loading of the data table, CICS reads the entire VSAM KSDS file and attempts to place all records in the data table. The XDTRD user exit can be used to limit the records that are placed in the data table (using screening criteria appropriate to the applications sharing the data). If all records that pass the screening criteria (if any) will not fit into the data table[1], a "table full" condition applies during the initial loading of the data table.

- **Records subsequently added from the source data set**. Records from a VSAM KSDS file can be added to a data table after the initial loading[2] if either (1) records would not fit into the data table because the table became full during initial loading or (2) the user exit had excluded records during initial loading but an excluded record was subsequently required by an application. If the data table had become full during initial loading, some record must be deleted from the data table for another record to be subsequently added from the source data set.

- **New records written to the data table.** New records can be added to a data table via the WRITE file control command. The XDTAD global user exit program can be used to limit the records that are added to the data table as a result of a WRITE request issued to a data table. If all records that pass the screening criteria (if any) will not fit into the data table[3], a "table full" condition is generated and the write is rejected.

Applications can reference the data table during initial loading. These references will produce a "record not found" condition if the references are to records outside the range of those already loaded into the data table. For both CICS-maintained data tables and user-maintained data tables, the "record not found" condition can occur during initial loading. These conditions should be small and can normally be ignored.

For CICS-maintained data tables, small numbers of the "record not found" condition after initial loading should not normally be a cause for alarm. However, a large number of "records not found" will cause unnecessary overhead and delay to applications[4].

---

[1] The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be placed in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that the entire VSAM KSDS file can be placed in the data table (subject, of course, to screening criteria applied by the XDTRD user exit).

[2] Records from the source data set cannot be added to a user-maintained data table after the initial loading of the data table. User-maintained data tables are separated from the source data set after loading.

[3] The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be added to in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that there is no limit to the number of records that can be added to the data table (subject, of course, to screening criteria applied by the XDTAD user exit).

[4] If the "record not found" condition does occur after initial loading of a user-maintained data table, either the data table is too small or the screening criteria in the XDTRD user exit does not match the record reference patterns of the application. These situations are analyzed in Rule CIC402.

For user-maintained data tables, the “record not found” condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the data table, any “record not found” condition for user-maintained data tables normally indicates a coding or logic error with the application.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses the A17DTRNF variable to assess the number of records not found situation. The A17DTRNF variable contains a count of the number of times CICS attempted to read a record but was unable to satisfy the read request because the record was not in the data table.

The CICS statistics contain an indicator (the A17DTTYP variable) describing whether a shared data table is CICS-maintained or user-maintained. Unfortunately, this indicator is set only when the VSAM data set is closed. CPExpert does a “reverse scan” of the shared data table statistics to identify CLOSED status of a shared data table. A17DTTYP is propagated to all observations prior to CLOSE, so CPExpert can identify whether a shared data table is CICS-maintained or User-maintained. Additionally, CPExpert examines all variables that represent data set accesses that modify the data set (these are the A17DSGU, A17DSWRU, A17DSWRA, A17DSDEL, A17RMDEL, and A17DSBRU variables). If the source data set is modified, the data table must be a CICS-maintained dat table.

This approach works only if the shared data table actually was closed within the CICS statistics data available in the performance data base being analyzed by CPExpert, or if modifications were made to the data set. If the shared data table CLOSE record is not present in the data or if modifications were made to the data set, CPExpert will be unable to determine whether the shared data table is CICS-maintained or user-maintained. From a practical matter, this should be of little import as the analysis depending on CICS-maintained versus user-maintained data tables would not be seriously effected for tables that are not closed and which have no source data set access.

CPExpert produces Rule CIC403 when the A17DTRNF value is greater than the **UMTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **UMTRNF** guidance variable is 0, indicating that CPExpert should produce Rule CIC403 when any read request resulted in a “record not found” condition.

Note that during initial loading of the data table, any read requests outside the range of records loaded will produce a count for the “record not found” condition. As mentioned earlier, this is not normally a cause for alarm. In an attempt to eliminate these read requests during initial table loading, CPExpert suppresses Rule CIC403 for any CICS interval statistics period when the table is being loaded (this suppression is based on the A17DTAVR variable being non-zero).

Suggestion: If Rule CIC403 is produced, you consider the following alternatives:

- Consult with applications personnel to review the application logic and coding. It is possible that an error exists with the application. Alternatively, the application might "test" for the presence of a record before adding a new record. If this is the case, the UMTRNF guidance should be changed for this specific table (Section 3 describes how to change guidance for individual shared data tables).

- Increase the number of records allowed in the shared data table. The number of records allowed in the shared data table is controlled by MAXNUMRECS parameter of the DEFINE FILE command. The DEFINE FILE command can be applied at the CEDA panel.

  Increasing the number of records in the shared data table would result in a larger table, using more virtual storage. Using more virtual storage could increase CICS paging operations, or increase overall system paging. Consequently, paging rates should be monitored if there is a large increase in the number of records allowed in the shared data table.

- If increasing the number of records allowed in the shared data table is not feasible, determine whether the XDTRD user exit has been used for the VSAM KSDS file. If the XDTRD user exit **has** been used, consider revising the screening criteria in the user exit to include a larger number of records during loading. The screening criteria might not match the records required by the application (perhaps because of different transactions or different distribution of transaction types requiring different records). Consequently, the existing screening criteria might exclude more records than are required by the applications or transaction patterns being submitted.

- You can change the UMTRNF guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC403 is produced too often.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
   Section 6.1 Using the DEFINE FILE command to define data tables
   Section 8.4 Interpreting data table statistics

Rule CIC404: High data set activity for CICS-maintained shared data table

Finding: The CICS interval statistics showed that there was high VSAM source data set activity for a CICS-maintained shared data table.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. With a CICS-maintained data table, any operation requiring access to the source data set reduces the efficiency of the data table.

CICS supports two types of data table :

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is *automatically applied* to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

For CICS-maintained data tables, the following file control commands access the VSAM source data set:

- READ commands with the UPDATE or RBA options.

- STARTBR, RESETBR, READNEXT, and READPREV commands with the RBA option.

- ENDBR command for a browse sequence that has accessed the source data set.

- READ and browse commands (that would usually access only the data table) that find a gap in the key sequence of records in the data table. This gap might indicate that one or more records are missing from the data table.

- READ, READNEXT, and READPREV commands for records that are currently being processed by a WRITE, REWRITE, or DELETE command. These commands need to first access the data table to determine that this situation exists.

- WRITE, REWRITE, and DELETE commands. These commands first update the source data set, and then attempt to update the data table.

- READ requests that encounter a "record not found" condition must access the VSAM source data set to retrieve the record not in the data table.

If a large percent of file control commands access the VSAM source data set, the benefits of using a data table can outweigh the overhead and virtual storage costs of maintaining a data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data set[1], using the following algorithm:

$$\textit{Percent VSAM source data set accesses} = \frac{\textit{VSAM data set accesses}}{\textit{All file accesses}}$$

where

VSAM data set accesses	=	A17DSGU+A17DSWRU+A17DSWRA+A17DSDEL + A17RMDEL + A17DSBRU - A17DTAVR
All file accesses	=	A17DSRD+A17DSGU+A17DSBR+A17DSWRU+ A17DSWRA+A17DSDEL+A17RMDEL+A17DSBRU- A17DTAVR

Please refer to the CICS Performance Guides for a description of the individual variables.

[1] Note that this algorithm would produce results only for CICS-maintained data tables. There would be no corresponding references to the VSAM source data set for user-maintained data tables.

Note that removing the A17DTAVR value from the accesses accounts for the initial table loading process.

Some CICS file control commands might reference both the data table and the VSAM source data set. These commands can create more overhead than if only the VSAM source data set were referenced. These file control commands are:

- READ and browse commands (that would usually access only the data table) that find a gap in the key sequence of records in the data table. IBM documentation lists reasons that one or more records might be missing from the data table:

- Records were been suppressed by a user exit during table loading.

- The maximum number of records (specified by the MAXNUMRECS value in the DEFINE FILE command) had been reached.

- Insufficient virtual storage was available for the data table.

- Some abnormal event has occurred.

- READ, READNEXT, and READPREV commands for records that were being processed by a WRITE, REWRITE, or DELETE command. These commands must access the data table to determine that this situation exists.

- WRITE, REWRITE, and DELETE commands. These commands are always executed in the File Owning Region (FOR), where they first update the source data set. If the update to the source data set is successful, a corresponding change to the data table is attempted using local shared data table services in the FOR. In the case of a WRITE command, the addition of the record to the data table might be rejected by the XDTAD user exit, or might fail because the data table is full, or insufficient virtual storage is available.

CPExpert produces Rule CIC404 when the percent VSAM source data set accesses is more than the value specified by the **DSPCTSRC** guidance variable in USOURCE(CICGUIDE). The default value for the **DSPCTSRC** is 25 indicating that CPExpert should produce Rule CIC404 whenever more than 25% of the file accesses required that CICS access the VSAM source data set.

Suggestion: For data tables to be effective, a significant percent of the file control commands must access only the data table rather than cause CICS to access the VSAM source data set. If Rule CIC404 is produced, you consider the following alternatives:

- You should consult with applications personnel to review the application logic and coding. It is possible that an error exists with the application.

- If the high access to the source data set is a natural result of the application's correct logic, perhaps the file is not suited for a data table. This conclusion would become more certain if a *high* percent of file accesses required that CICS access the VSAM source data set. For example, if more than 75% of the file accesses[2] required that CICS access the VSAM source data set, the file probably should not be included in a data table.

- You can change the DSPCTSRC guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC404 is produced too often.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

[2] Rule CIC406 will be produced if more than 75% of the file accesses required that CICS access the VSAM source data set.

Rule CIC405: High data set activity for CICS-maintained shared data table

Finding: The CICS interval statistics showed that there was high VSAM source data set activity for a CICS-maintained shared data table.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. With a CICS-maintained data table, any operation requiring access to the source data set reduces the efficiency of the data table.

CICS supports two types of data table :

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is *automatically applied* to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

For CICS-maintained data tables, the following file control commands access the VSAM source data set:

- READ commands with the UPDATE or RBA options.

- STARTBR, RESETBR, READNEXT, and READPREV commands with the RBA option.

- ENDBR command for a browse sequence that has accessed the source data set.

- READ and BROWSE commands (that would usually access only the data table) that find a gap in the key sequence of records in the data table. This gap might indicate that one or more records are missing from the data table. IBM documentation lists reasons that one or more records might be missing from the data table:

  - Records had been suppressed by the XDTRD global exit during table loading or by the XDTAD global exit when records were added to the table.

  - The maximum number of records (specified by the MAXNUMRECS value in the DEFINE FILE command) had been reached.

  - Insufficient virtual storage was available for the data table.

  - Some abnormal event occurred that prevented a record from being in the data table.

- READ, READNEXT, and READPREV commands for records that are currently being processed by a WRITE, REWRITE, or DELETE command. These commands need to first access the data table to determine that this situation exists.

- WRITE, REWRITE, and DELETE commands. These commands first update the source data set, and then attempt to update the data table.

- READ requests that encounter a "record not found" condition must access the VSAM source data set to retrieve the record not in the data table.

If a large percent of file control commands access the VSAM source data set, the benefits of using a data table can outweigh the overhead and virtual storage costs of maintaining a data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data set[1], using the following algorithm:

$$\textit{Percent VSAM source data set accesses} = \frac{\textit{VSAM data set accesses}}{\textit{All file accesses}}$$

[1]Note that this algorithm would produce results only for CICS-maintained data tables. There would be no corresponding references to the VSAM source data set for user-maintained data tables.

where

VSAM data set accesses	=	A17DSGU+A17DSWRU+A17DSWRA+A17DSDEL + A17RMDEL + A17DSBRU - A17DTAVR
All file accesses	=	A17DSRD+A17DSGU+A17DSBR+A17DSWRU+ A17DSWRA+A17DSDEL+A17RMDEL+A17DSBRU- A17DTAVR

Please refer to the CICS Performance Guides for a description of the individual variables.

Note that removing the A17DTAVR value from the accesses accounts for the initial table loading process.

CPExpert produces Rule CIC405 when the percent VSAM source data set accesses is more than the value specified by the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTDTSRC** is 50 indicating that CPExpert should produce Rule CIC405 whenever more than 50% of the file accesses required that CICS access the VSAM source data set.

Before producing Rule CIC405, however, CPExpert applies the **MINSDTIO** variable to ensure that Rule CIC405 is produced only for shared data tables with a reasonably high amount to of I/O activity. The default value for the MINSDTIO is 500, indicating that Rule CIC405 will be suppressed unless at least 500 VSAM file control commands are issued to the shared data table.

Suggestion: For data tables to be effective, a significant percent of the file control commands must access only the data table rather than cause CICS to access the VSAM source data set. If Rule CIC405 is produced regularly, you consider the following alternatives:

- Determine whether records were not found in the shared data table, requiring access to the VSAM source data set. Rule CIC402 would normally be produced if records were not found in the shared data table. Records might not be found because number of records was limited by the MAXNUMRECS variable, because records were excluded by the XDTRD global exit while the table was loaded, or because records were excluded by the XDTAD global exit while records were dynamically added to the table. If records were not found, follow the advice given by Rule CIC402.

- You should consult with applications personnel to determine whether the VSAM file control commands can be modified to use less restrictive access techniques.

- You should consult with applications personnel to review the application logic and coding. It is possible that an error exists with the application.

- If the high access to the source data set is a natural result of the application's correct logic, perhaps the file is not suited for a data table. This conclusion would become more certain if a *high* percent of file accesses required that CICS access the VSAM source data set. For example, if more than 90% of the file accesses[2] required that CICS access the VSAM source data set, the file probably should not be included in a data table.

- You can change the **PCTDTSRC** guidance variable in USOURCE(CICGUIDE) if you believe that Rule CIC405 is produced too often.

- You can change the **MINSDTIO** guidance variable in USOURCE(CICGUIDE) if you want to have the CIC405 logic applied only for shared data tables with a larger number of I/O accesses.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

[2] Rule CIC406 will be produced if more than 90% of the file accesses required that CICS access the VSAM source data set.

Rule CIC406: VSAM data set might not be good candidate for shared data table

Finding: The CICS interval statistics showed that there was *very* high VSAM source data set activity for a CICS-maintained shared data table.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the CICS interval statistics.

Discussion: An application can specify that a CICS VSAM key-sequenced data set (KSDS) file is to use shared data table services. When the file is opened, this specification causes CICS to copy the contents of the file into an *MVS data space*. The records can be accessed in an MVS data space significantly quicker than records read from the VSAM data set or via reads serviced by a Local Shared Resources (LSR) pool.

Since a major benefit of a data table is that records in the data table can be accessed quickly, this benefit is available only if the records actually are accessed in the data table rather than in the VSAM source data set. With a CICS-maintained data table, any operation requiring access to the source data set reduces the efficiency of the data table.

CICS supports two types of data table :

- **CICS-maintained data tables**. A CICS-maintained data table is one that CICS keeps in synchronization with their source data sets. That is, any update or delete action on a record in the data table is *automatically applied* to the source data set *before* being applied to the data table.

- **User-maintained data tables**. A user-maintained data table (UMT) is one that is not maintained by CICS, but is completely maintained by user code. A UMT is detached from its source data set after the table is loaded from the source data set, and changes made to the UMT are *not* reflected in the VSAM source data set.

For CICS-maintained data tables, the following file control commands access the VSAM source data set:

- READ commands with the UPDATE or RBA options.

- STARTBR, RESETBR, READNEXT, and READPREV commands with the RBA option.

- ENDBR command for a browse sequence that has accessed the source data set.

- READ and BROWSE commands (that would usually access only the data table) that find a gap in the key sequence of records in the data table. This gap might indicate that one or more records are missing from the data table. IBM documentation lists reasons that one or more records might be missing from the data table:

  - Records had been suppressed by the XDTRD global exit during table loading or by the XDTAD global exit when records were added to the table.

  - The maximum number of records (specified by the MAXNUMRECS value in the DEFINE FILE command) had been reached.

  - Insufficient virtual storage was available for the data table.

  - Some abnormal event occurred that prevented a record from being in the data table.

- READ, READNEXT, and READPREV commands for records that are currently being processed by a WRITE, REWRITE, or DELETE command. These commands need to first access the data table to determine that this situation exists.

- WRITE, REWRITE, and DELETE commands. These commands first update the source data set, and then attempt to update the data table.

- READ requests that encounter a "record not found" condition must access the VSAM source data set to retrieve the record not in the data table.

If a large percent of file control commands access the VSAM source data set, the benefits of using a data table can outweigh the overhead and virtual storage costs of maintaining a data table.

Shared data table statistics are available in MXG file CICFCR. CPExpert uses data in CICFCR to calculate the percent of file control commands that accessed the VSAM source data set[1], using the following algorithm:

$$\textit{Percent VSAM source data set accesses} = \frac{\textit{VSAM data set accesses}}{\textit{All file accesses}}$$

[1] Note that this algorithm would produce results only for CICS-maintained data tables. There would be no corresponding references to the VSAM source data set for user-maintained data tables.

where

VSAM data set accesses	=	A17DSGU+A17DSWRU+A17DSWRA+A17DSDEL + A17RMDEL + A17DSBRU - A17DTAVR
All file accesses	=	A17DSRD+A17DSGU+A17DSBR+A17DSWRU+ A17DSWRA+A17DSDEL+A17RMDEL+A17DSBRU- A17DTAVR

Please refer to the CICS Performance Guides for a description of the individual variables.

Note that removing the A17DTAVR value from the accesses accounts for the initial table loading process.

CPExpert produces Rule CIC406 when **more than 90%** of file control commands accessed the VSAM source data set, and this situation existed for **more than 75%** of the CICS statistics intervals being analyzed.

Before producing Rule CIC406, however, CPExpert applies the **MINSDTIO** variable to ensure that Rule CIC406 is produced only for shared data tables with a reasonably high amount to of I/O activity. The default value for the MINSDTIO is 500, indicating that Rule CIC406 will be suppressed unless at least 500 VSAM file control commands are issued to the shared data table.

Suggestion: For data tables to be effective, a significant percent of the file control commands must access only the data table rather than cause CICS to access the VSAM source data set. If Rule CIC406 is produced regularly, you consider the following alternatives:

- Determine whether records were not found in the shared data table, requiring access to the VSAM source data set. Rule CIC402 would normally be produced if records were not found in the shared data table. Records might not be found because number of records was limited by the MAXNUMRECS variable, because records were excluded by the XDTRD global exit while the table was loaded, or because records were excluded by the XDTAD global exit while records were dynamically added to the table. If records were not found, follow the advice given by Rule CIC402.

- You should consult with applications personnel to determine whether the VSAM file control commands can be modified to use less restrictive access techniques.

- You should consult with applications personnel to review the application logic and coding. It is possible that an error exists with the application.

- If the high access to the source data set is a natural result of the application's correct logic, perhaps the file is not suited for a data table. This conclusion would become more certain if a *high* percent of file accesses required that CICS access the VSAM source data set. If more than 90% of the file accesses required that CICS access the VSAM source data set, the file probably should not be included in a data table.

- You can change the **MINSDTIO** guidance variable in USOURCE(CICGUIDE) if you want to have the CIC406 logic applied only for shared data tables with a larger number of I/O accesses.

Reference: CICS/TS for OS/390 Release 1.3 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.1 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

CICS/TS for z/OS Release 2.2 *CICS Shared Data Tables Guide*:
Section 6.1 Using the DEFINE FILE command to define data tables
Section 8.4 Interpreting data table statistics

Rule CIC420: A large percent of CFDT structure list entries were in use

Finding: The CICS Coupling Facility Data Table (CFDT) Pool Server statistics showed that a large percent of CFDT structure list entries were in use for the coupling facility structure containing the CFDT pool.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or for z/OS.

Discussion: The CICS Coupling Facility Data Tables support provide a significant enhancement to shared data tables in a parallel sysplex. The CFDT design provides an excellent way to share file data using CICS file control, without resorting to VSAM record level sharing (RLS). The CFDT design eliminates the requirement for having a File Owning Region (as is required with normal shared data tables).

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

Access to a CFDT is via a *CFDT pool server* that supports a specific named pool. A CFDT pool server is started in an MVS image by starting a pool server region (as either a batch job or a started task) for each CFDT pool. Starting the pool server region invokes the pool server region program, DFHCFMN, which resides in an APF-authorized library. Each CDFT pool server provides access to only one CFDT pool, so there must be multiple CFDT pool servers if there are multiple CFDT pools.

A variety of parameters are provided to the CFDT pool server, to specify the name of the CFDT pool, list structure attributes, tuning parameters, various thresholds for warning and automatic server actions, etc. These parameters (other than the pool name, which has a default *prefix* of DFHCF) have default values, but the defaults can be changed based on user-specific requirements.

The application tasks issue standard CICS file control commands to read, write, browse, or delete CFDT records (READ, READNEXT, READPREV, WRITE, STARTBR, and DELETE). The pool server is responsible for translating these application task commands into XES list structure interfaces.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With CFDT pools, the list structure is the named pool, while the lists themselves are CFDTs within the named pool. The CFDT pool server designates the maximum number of lists (or tables) the CFDT pool is to have, and allocates the list structure based on parameters that are provided to the CFDT pool server.

A list entry consists of list entry controls and can optionally include an *adjunct area*, a *data entry*, or both. Data entries are composed of units of storage called *data elements*. In a coupling facility of CFLEVEL=0, data entries can be composed of 0 to 16 data elements. In a coupling facility of CFLEVEL=1 or higher, data entries can be composed of 0 to 255 data elements. In either case, a data entry could contain up to 64K (65536 bytes) of data.

When the list structure is allocated, XES establishes the number of data elements that are associated with data entries. This division of storage is referred to as the “entry-to-element ratio”.

The structure *alter* function provides for the expansion or contraction of the size of a structure, the reapportionment of the entry-to-element ratio of the structure's storage, and the alteration of the percentage of structure storage set aside for event monitor controls (EMCs). The structure alter processing is done either by using the IXLALTER macro or by issuing the SETXCF START,ALTER command. The IXLALTER macro allows an authorized user to request a change to the structure's size, the entry-to-element ratio, and the percentage of storage allocated for EMCs.

Starting with OS/390 Release 10, a structure can be *automatically* altered when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both.

With CFDT pools, the pool server monitors the total number of data entries and data elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the CFDT pool server *warning parameters*, a warning message (DFHCF0411 or DFHCF0412, for entries and elements, respectively) is issued. The warning message is repeated each time the number in use increases beyond further thresholds.

Each time the warning is issued, the CFDT pool server tests whether an automatic ALTER for the entry to element ratio should be performed. The test is done based on the CFDT pool server *automatic structure alter parameters*. This is based on the ratio between the current numbers of elements and entries actually in use.

IBM suggests that no more than 75% of the structure be used, to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

Section 2.10.8.4.2 (Approximate storage calculations) of the CICS/TS *System Definition Guide* provides calculations that can be used to calculate initial sizing of the storage for a CFDT pool. However, this algorithm might not yield an adequate structure size for some environments.

Further, the default element to entry ratio in the CFDT pool server *tuning parameters* is a simple 1:1, which might not be optimum for any particular CFDT environment.

Consequently, CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used.

CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the structure list entries that had been used, using the following algorithm:

$$\textit{Maximum percent structure list entries used} = \frac{S6ENTRHI}{S6ENTRMX}$$

where S6ENTRHI = Maximum number of list entries used since last reset
S6ENTRMX = Total list entries in the currently allocated structure

CPExpert produces Rule CIC420 when the maximum percent structure list entries used is more than the value specified by the **CFPCTENT** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTENT** is 70 indicating that CPExpert should produce Rule CIC420 whenever more than 70% of the list entries had been used.

Suggestion: Rule CIC420 indicates that the CFDT pool server either is exercising automatic alter algorithms, or is likely to exercise these algorithms as the thresholds in the *automatic structure alter parameters* are reached. If this finding is produced often, you should consider the following alternatives:

- Increase the initial amount of structure space that is available for the CFDT pool identified by this finding. Increasing the amount of initial structure space can be accomplished by increasing the INITSIZE (so more structure space is initially available). If more structure space is initially available, more entries and elements will be available and there is less probability that there will be a shortage of list entries.

- Increase the amount of storage allocated for the *maximum size* specified in the coupling facility resource management policy for the CFDT pool identified by this finding. This action normally should be taken only if the structure size has approached the maximum size specified. Be aware that Rule CIC425 will be produced if the structure runs out of space, but frequent occurrence of Rule CIC420 might indicate a pending "no space" condition.

- You could change the CFPCTENT guidance variable in USOURCE(CICGUIDE) so Rule CIC420 is produced less often. This action is not recommended, however since you should be aware of the potential problems (it is particularly important to be aware of <u>pending</u> problems) revealed by Rule CIC420

.

Reference: CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 2.10.8 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 27 (Starting a CFDT server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Section 4.3 (Starting a CFDT server)

Rule CIC421: A large percent of CFDT structure data elements were in use

Finding: The CICS Coupling Facility Data Table (CFDT) Pool Server statistics showed that a large percent of CFDT structure data elements had been used in the coupling facility structure containing the CFDT pool.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390, Release 1.3, or for z/OS.

Discussion: The CICS Coupling Facility Data Tables provide a significant enhancement to shared data tables in a parallel sysplex. The CFDT design provides an excellent way to share file data using CICS file control, without resorting to VSAM record level sharing (RLS). The CFDT design eliminates the requirement for having a File Owning Region (as is required with normal shared data tables).

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

Please refer to Rule CIC420 for a brief discussion of CFDT pools.

The structure *alter* function provides for the expansion or contraction of the size of a structure, the reapportionment of the entry-to-element ratio of the structure's storage, and the alteration of the percentage of structure storage set aside for event monitor controls (EMCs). The structure alter processing is done either by using the IXLALTER macro or by issuing the SETXCF START,ALTER command. The IXLALTER macro allows an authorized user to request a change to the structure's size, the entry-to-element ratio, and the percentage of storage allocated for EMCs.

Starting with OS/390 Release 10, a structure can be automatically altered when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both.

With CFDT pools, the pool server monitors the total number of entries and elements in use in the structure, using information returned by the coupling facility on every request. When the numbers in use exceed thresholds specified by the CFDT pool server *warning parameters*, a warning message (DFHCF0411 or DFHCF0412, for entries and elements, respectively) is issued. The warning message is repeated each time the number in use increases beyond further thresholds.

Each time the warning is issued, the CFDT pool server tests whether an automatic ALTER for the entry to element ratio should be performed. The test is done based on the CFDT pool server *automatic structure alter parameters*. This is based on the ratio between the current numbers of elements and entries actually in use.

IBM suggests that no more than 75% of the structure be used to minimize the risk of the structure becoming full, to avoid triggering low space warning messages, and to avoid additional activity required to alter entry to element ratios. However, the default ELEMENTWARN and ENTRYWARN warning parameters have a default value of **80**, which specify that warnings and automatic ALTER actions should be first triggered when 80% of the elements or entries are used.

Section 2.10.8.4.2 (Approximate storage calculations) of the CICS/TS *System Definition Guide* provides calculations that can be used to calculate initial sizing of the storage. However, this algorithm might not yield an adequate structure size for some environments.

Further, the default element to entry ratio in the CFDT pool server *tuning parameters* is a simple 1:1, which might not be optimum for your coupling facility data table environment.

Consequently, CPExpert provides an earlier warning of structure element and entry shortage by analyzing the maximum number of elements and entries that were used.

CFDT pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the maximum percent of the data elements that had been used, using the following algorithm:

$$\textit{Maximum percent data elements used} = \frac{S6ELEMHI}{S6ELEMMX}$$

where S6ELEMHI = Maximum number of data elements used since last reset
S6ELEMMX = Total data elements in the currently allocated structure

CPExpert produces Rule CIC421 when the maximum percent data elements used is more than the value specified by the **CFPCTELE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTELE** is 70 indicating that CPExpert should produce Rule CIC421 whenever more than 70% of the data elements had been used.

Suggestion: Rule CIC421 indicates that the CFDT pool server either is exercising automatic alter algorithms, or is likely to exercise these algorithms as the thresholds in the *automatic structure alter parameters* are reached. If this finding is produced often, you should consider the following alternatives:

- Increase the initial amount of structure space that is available for the CFDT pool identified by this finding. Increasing the amount of initial structure space can be accomplished by increasing the INITSIZE (so more structure space is initially available). If more structure space is initially available, more entries and elements will be available and there is less probability that there will be a shortage of data elements.

- Increase the amount of storage allocated for the *maximum size* specified in the coupling facility resource management policy for the CFDT pool identified by this finding. This action normally should be taken only if the structure size has approached the maximum size specified. Be aware that Rule CIC425 will be produced if the structure runs out of space, but frequent occurrence of Rule CIC421 might indicate a pending “no space” condition.

- You could change the CFPCTELE guidance variable in USOURCE(CICGUIDE) so Rule CIC421 is produced less often. This action is not recommended, however since you should be aware of the potential problems (it is particularly important to be aware of <u>pending</u> problems) revealed by Rule CIC421.

Reference: CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 2.10.8 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 27 (Starting a CFDT server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Section 4.3 (Starting a CFDT server)

Rule CIC422: CFDT structure entry data was larger than input buffer length

Finding: The CICS Coupling Facility Data Table (CFDT) Pool Server statistics showed that CFDT structure entry data was larger than input buffer length.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or for z/OS.

Discussion: A Coupling Facility Data Table is kept in a *named Pool* in an MVS coupling facility. A CFDT Pool consists of an XES list structure, which is accessed through a cross-memory queue server region. There can be multiple CFDT pools, each containing one or more CFDTs.

CFDT Pool server statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to calculate the percent of requests that timed out, using the following algorithm:

$$\textit{Percent requests that timed out} = \frac{\textit{S6RSP2CT}}{\textit{S6RSP1CT} + \textit{Abnormal responses}}$$

where S6RSP1CT = Number of normal responses
S6RSP2CT = Number of requests that timed out
Abnormal = S6RSP2CT + S6RSP3CT + S6RSP4CT + S6RSP5CT+ S6RSP6CT + S6RSP7CT + S6RSP8CT

S6RSP3CT = Specified entry not found
S6RSP4CT = Version check failed for entry being updated
S6RSP5CT = List authority comparison failed
S6RSP6CT = Maximum list key reached
S6RSP7CT = List structure was out of space
S6RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC422 when the percent requests that timed out is greater than the value specified by the **CFPCTTIM** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTTIM** is 1 indicating that CPExpert should produce Rule CIC422 whenever more than one percent of the requests timed out because the request exceeded the coupling facility model-dependent time-out criteria.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of CFDT pools to determine whether the IXLLIST commands can be issued with fewer actions against temporary storage entries. For example, the DELETE command might be issued more frequently to delete a smaller number of entries with each delete.

- Change the CFPCTTIM guidance variable in USOURCE(CICGUIDE) so Rule CIC422 is produced only when you wish to be aware of applications that cause a larger percent of coupling facility time-outs.

- You can specify **%LET CFPCTTIM = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent requests timed-out cannot be greater than 100), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: z/OS V1R1 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R2 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R3 *MVS Programming: Sysplex Services Reference*
Section 42.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

z/OS V1R4 *MVS Programming: Sysplex Services Reference*
Section 46.0 IXLLIST -- List Services and following sections related to the list structure commands shown on Page 1 of this Rule Description

Rule CIC423: High percent entries (either table or item) not found in CFDT

Finding: The CICS Coupling Facility Data Table (CFDT) list structure statistics showed that a large percent of entries (either table or item) were not found.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or for z/OS.

Discussion: The CICS Coupling Facility Data Tables provide a significant enhancement to shared data tables in a parallel sysplex. The CFDT design provides an excellent way to share file data using CICS file control, without resorting to VSAM record level sharing (RLS). The CFDT design eliminates the requirement for having a File Owning Region (as is required with normal shared data tables).

CICS CFDT support is designed to provide sharing of working data within a sysplex, while maintaining update integrity of the data. The working data is held in a coupling facility data table, which is contained in a *named pool* located in coupling facility list structure. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a list structure in a coupling facility.

Access to a CFDT is via a *CFDT pool server* that supports a specific named pool. A CFDT pool server is started in an MVS image by starting a pool server region (as either a batch job or a started task) for each CFDT pool. Starting the pool server region invokes the pool server region program, DFHCFMN, which resides in an APF-authorized library. Each CFDT pool server provides access to only one CFDT pool, so there must be multiple CFDT pool servers if there are multiple CFDT pools.

Applications access data in a CFDT using standard file control commands (read, write, delete, etc.).

Records are placed into a CFDT in one of two ways: (1) records placed in the CFDT by the initial loading of the CFDT from the source data set (if a source data set is defined), and (2) new records written to the CFDT after the CFDT has been loaded.

- **Initial loading of the CFDT**. Unlike a shared data table, a CFDT does not require a source data set. A CFDT file definition can specify that there is no associated source data set, allowing an empty CFDT to be created. Alternatively, a CFDT may have a source data set (just as with a data table), and the CFDT is loaded when the source data set is first opened. If initial loading of the CFDT occurs, CICS reads the entire VSAM KSDS file and attempts to place all records in the CFDT. The XDTRD user exit can be used to limit the records that are placed in the CFDT (using screening criteria appropriate to the applications sharing the data). If all records that pass the screening criteria (if any) will not fit into the CFDT[1], a "table full" condition applies during the initial loading of the CFDT.

- **New records written to the CFDT.** New records can be added to a CFDT via the WRITE file control command. The XDTAD global user exit program can be used to limit the records that are added to the CFDT as a result of a WRITE request issued to the CFDT. If all records that pass the screening criteria (if any) will not fit into the CFDT[2], a "table full" condition is generated and the write is rejected.

Applications can reference the CFDT during initial loading. These references will produce a "record not found" condition if the references are to records outside the range of those already loaded into the data table. These conditions should be small and can normally be ignored.

The "record not found" condition should not occur after loading of the data table. Since the VSAM source data set is not available after the initial loading of the CFDT (if a source data set existed), any "record not found" condition for normally indicates (1) a coding or logic error with the application, (2) an attempt to reference a record that should be in the CFDT (but was not in the CFDT because, for example, the MAXNUMRECS had been reached or the record had been suppressed by one of the user exits mentioned above), or (3) a deliberate reference to determine whether the record exists.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP3CT variable contains a count of the number of times CICS attempted to read a record from the CFDT, but was unable to satisfy the read request because the record was not in the CFDT.

[1]The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be placed in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that the entire VSAM KSDS file can be placed in the CFDT (subject, of course, to screening criteria applied by the XDTRD user exit).

[2]The MAXNUMRECS parameter of the DEFINE FILE command can used to limit the number of records that can be added to in the data table. The MAXNUMRECS parameter has a default maximum of NOLIMIT, which means that there is no limit to the number of records that can be added to the data table (subject, of course, to screening criteria applied by the XDTAD user exit).

CPExpert calculates the percent of requests that encountered a "specified entry (table or item) not found" condition, using the following algorithm:

$$Percent\ entries\ not\ found = \frac{S6RSP3CT}{S6RSP1CT + Abnormal\ responses}$$

where S6RSP1CT = Number of normal responses
Abnormal = S6RSP2CT + S6RSP3CT + S6RSP4CT + S6RSP5CT+ S6RSP6CT + S6RSP7CT + S6RSP8CT

S6RSP2CT = Entry data was larger than the input buffer length
S6RSP3CT = Specified entry (table or item) not found
S6RSP4CT = Version check failed for entry being updated
S6RSP5CT = List authority comparison failed
S6RSP6CT = A table reached maximum number of items
S6RSP7CT = List structure became full
S6RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC423 when the percent requests that resulted in an "entry (table or item) not found" condition is greater than the **CFPCTRNF** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTRNF** guidance variable is 0.1, indicating that CPExpert should produce Rule CIC423 when more than 0.1% of the requests resulted in an "entry (table or item) not found" condition.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Under normal conditions, Rule CIC402 or Rule CIC403 should have been produced to identify the shared data table to which this rule (Rule CIC423) applies. Examine the applications using the shared data table to determine why the "specified entry (table or item) was not found" condition is encountered in the CFDT. With many situations, this condition should be considered a program error and should be remedied.

- As mentioned earlier, there *are* situations in which some number of "specified entry (table or item) was not found. As examples, (1) an attempt to reference a record that should be in the CFDT (but was not in the CFDT because, for example, the MAXNUMRECS had been reached or the record had been suppressed by one of the user exits mentioned above), or (2) a deliberate reference was made to determine whether the record exists. If these situations exist, the CFPCTRNF guidance variable should be changed.

- Change the **CFPCTRNF** guidance variable in USOURCE(CICGUIDE) so Rule CIC423 is produced only when you wish to be aware of a larger percent of "specified entry (table or item) was not found" condition.

- You can specify **%LET CFPCTRNF = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent requests encountering entries not found in CFDT) cannot be greater than 100), or you can "turn off" the rule using the process described in Section 3 of this User Manual.

Reference: CICS/TS for OS/390 Release 1.3

CICS System Definition Guide: Section 2.10.8 (Coupling facility data tables)

CICS Customization Guide: Section 1.1.7 (Data tables management exits XDTRD, XDTAD, and XDTLC)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Chapter 27 (Starting a CFDT server)

CICS Customization Guide: Chapter 23 (Data tables management exits XDTRD, XDTAD, and XDTLC)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3 (Starting a CFDT server)

CICS Customization Guide: Section 1.1.7 (Data tables management exits XDTRD, XDTAD, and XDTLC)

Rule CIC424: Maximum number of records was reached for CFDT

Finding: The CICS Coupling Facility Data Table (CFDT) list structure statistics showed that the maximum number of records had been reached for the CFDT.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The level of impact depends on whether the situation was expected (and MAXNUMRECS value had been specified to restrict the number of records) or the situation was unexpected (and expected records were not in the CFDT).

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or for z/OS.

Discussion: Coupling Facility Data Tables are kept in a *named pool* in an MVS coupling facility. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a *list structure* in a coupling facility.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With CFDT pools, the list structure is the CFDT pool, while the lists themselves are CFDTs within the CFDT pool. The CFDT pool server allocates the list structure based on parameters that are provided to the CFDT pool server at startup. These parameters allow a user to specify controls in such areas as allocation, security, tuning, reserved space, etc. Two parameters control the size of the CFDT pool:

- The MAXTABLES parameter provided to the CFDT pool server specifies how many tables can be in the CFDT pool list structure. The default value for the MAXTABLES parameter is 1000, indicating that 1000 tables can be opened in the CFDT pool.

- The POOLSIZE parameter specifies the initial amount of coupling facility storage to be allocated for the pool list structure. IBM recommends that a value not be specified for the POOLSIZE parameter, which allows the server to obtain an initial allocation using the parameters specified in the Coupling Facility Resource Management (CFRM) policy.

CICS automatically creates a CFDT within a CFDT pool when the first reference to the associated VSAM (KSDS) data set requires the CFDT to be opened. Once the CFDT is created, CICS can optionally load the coupling facility data table automatically from a source VSAM data set when it is opened, or the file definition can specify that there is no associated source data set, allowing an empty CFDT to be created.

The XDTRD global user exit can be used to limit the records that are placed in the CFDT during *initial loading* of the CFDT. The XDTAD global user exit program can be used to limit the records that are *added* to the CFDT as a result of a *WRITE request* issued to a data table. Both these user exits can use screening criteria appropriate to the applications sharing the data.

The MAXNUMRECS parameter provided with the VSAM file definition specifies the maximum number of records that can be in the CFDT itself. The default value for the MAXNUMRECS parameter is NOLIMIT, indicating that there is no limit on the maximum number of records.

If a value is specified for the MAXNUMRECS parameter, the CFDT can reach “full” status. When the number of records in the CFDT reaches the MAXNUMRECS value, the CFDT is marked “full” and no further records can be added to the CFDT until records are deleted to make space. The WRITE request is rejected, and the transaction must handle the NOSPACE return with an exception handling routine.

IBM suggests that for a *recoverable* coupling facility data table, the MAXNUMRECS value should be between 5% and 10% more than the maximum number of records that the table is expected to contain. This increased MAXNUMRECS value[1] allows for additional records that might be created internally for processing recoverable requests.

If this margin is not specified, the NOSPACE condition can be raised on a WRITE or REWRITE request to a recoverable coupling facility data table that apparently has fewer records than the MAXNUMRECS limit specifies.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. The S6RSP6CT variable contains a count of the number of times that a list became full. CPExpert calculates the percent of requests that encountered a “list full” condition, using the following algorithm:

$$\mathit{Percent\ list\ full} = \frac{\mathit{S6RSP6CT}}{\mathit{S6RSP1CT} + \mathit{Abnormal\ responses}}$$

[1] The number of additional records required for this internal processing depends on the level of use of the coupling facility data table, and the nature of that use.

where S6RSP1CT = Number of normal responses
Abnormal = S6RSP2CT + S6RSP3CT + S6RSP4CT + S6RSP5CT+ S6RSP6CT + S6RSP7CT + S6RSP8CT

S6RSP2CT = Entry data was larger than the input buffer length
S6RSP3CT = Specified entry (table or item) not found
S6RSP4CT = Version check failed for entry being updated
S6RSP5CT = List authority comparison failed
S6RSP6CT = List Full (a table reached maximum number of items)
S6RSP7CT = List structure became full
S6RSP8CT = Other IXLLIST return code occurred

CPExpert produces Rule CIC424 when the percent of List Full conditions is greater than the value specified by the **CFPCTFUL** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTFUL** is 0, indicating that CPExpert should produce Rule CIC424 whenever any List Full conditions occurred.

Suggestion: If this finding is produced, you should assess whether reaching the MAXNUMRECS value is expected.

- The CFDT list structure statistics relate to the entire CFDT pool, and do not identify a specific CFDT to which the statistics apply. From this view, the CFDT list structure is a “summary” of conditions for all tables in the coupling facility list structure.

  However, Rule CIC401 should have been produced to identify the shared data table to which this rule (Rule CIC424) applies. Consequently, you can use the information provided by Rule CIC401 to identify the specific shared data tables that reached MAXNUMRECS value. Examine the applications using the shared data table to determine why the “List Full” condition was encountered in the CFDT.

- It is quite likely that the MAXNUMRECS value was specified without considering the margin required to allow for additional records that might be created internally for processing recoverable requests. If this is the case, the MAXNUMRECS should be increased to account for these additional records.

- Reaching the List Full condition might be an expected result of the value specified for MAXNUMRECS. In this case, you should either change the CFPCTFUL guidance variable in USOURCE(CICGUIDE) so that Rule CIC424 would be produced only with a large number of List Full situations, or “turn off” Rule CIC424.

- Reaching MAXNUMRECS might **not** be an expected result, if there is more than one occurrence of the situation. The MAXNUMRECS value could have been set to curtail the number of records being read from the source data set. In this case, you should consider the following:

  - The user exit XDTRD might be used to screen the records being loaded into the CFDT. Applying a scheme to screen the records loaded into the CFDT might be more suitable than arbitrarily restricting the number of records loaded.

  - The user exit XDTAD might be used to screen the records being written to the CFDT after loading has completed (or after the null CFDT was established, in the case of a CFDT that has not corresponding source data set). Applying a scheme to screen the records written to the CFDT might be more suitable than arbitrarily restricting the number of records loaded.

  - Multiple occurrences of the List Full condition might imply that there is an error in the application logic. A List Full condition can be removed only after records are deleted from the CFDT. If records are deleted, additional records could be written to the CFDT raising the number of records to the limit set by MAXNUMRECS and creating a new List Full condition.

    More than one List Full condition implies that this cycle was executed, and the larger the number of List Full conditions implies a corresponding larger number of times the cycle was executed. A looping transaction (or a transaction with incorrect logic) could generate a very large number of List Full conditions, with corresponding overhead and performance degradation.

- You can change the CFPCTFUL guidance variable in USOURCE(CICGUIDE) so Rule CIC424 is produced only when you wish to be aware of a larger number of List Full situations.

- You can “turn off” the rule using the process described in Section 3 of this User Manual. However, this alternative is **not** recommended! You should always be aware of List Full situations.

Reference: CICS/TS for OS/390 Release 1.3

CICS System Definition Guide: Section 2.10.8 (Coupling facility data tables)

CICS Customization Guide: Section 1.1.7 (Data tables management exits XDTRD, XDTAD, and XDTLC)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Chapter 27 (Starting a CFDT server)

CICS Customization Guide: Chapter 23 (Data tables management exits XDTRD, XDTAD, and XDTLC)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3 (Starting a CFDT server)

CICS Customization Guide: Section 1.1.7 (Data tables management exits XDTRD, XDTAD, and XDTLC)

Rule CIC425: Coupling Facility Data Table list structure was out of space

Finding: The CICS Coupling Facility Data Table (CFDT) list structure statistics showed that the CFDT pool list structure was out of space.

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or for z/OS.

Discussion: Coupling Facility Data Tables are kept in a *named pool* in an MVS coupling facility. There can be multiple CFDT pools, each containing one or more CFDTs. Each CFDT pool is defined, using MVS cross-system extended services (XES), as a *list structure* in a coupling facility.

A list structure consists of a set of lists and an optional lock table of exclusive locks (which can be used to serialize the use of lists, list entries, or other resources in the list structure). Each list is pointed to by a *list header* and can contain a number of *list entries*. With CFDT pools, the list structure is the CFDT pool, while the lists themselves are CFDTs within the CFDT pool.

The CFDT pool server allocates the list structure based on parameters that are provided to the CFDT pool server at startup. These parameters allow a user to specify controls in such areas as allocation, security, tuning, reserved space, etc. Two parameters control the size of the CFDT pool:

- The MAXTABLES parameter provided to the CFDT pool server specifies how many tables can be in the CFDT pool list structure. The default value for the MAXTABLES parameter is 1000, indicating that 1000 tables can be opened in the CFDT pool.

- The POOLSIZE parameter specifies the initial amount of coupling facility storage to be allocated for the pool list structure. IBM recommends that a value not be specified for the POOLSIZE parameter, which allows the server to obtain an initial allocation using the parameters specified in the Coupling Facility Resource Management (CFRM) policy.

CICS automatically creates a CFDT within the CFDT pool when the first reference to the associated VSAM (KSDS) data set requires the CFDT to be opened. Once the CFDT is created, CICS can optionally load the coupling facility data table automatically from a source VSAM data set when it is opened, or the file definition can specify that there is no associated source data set, allowing an empty CFDT to be created.

The XDTRD global user exit can be used to limit the records that are placed in the CFDT during *initial loading* of the CFDT. The XDTAD global user exit program can be used to limit the records that are *added to the CFDT as a result of a WRITE request* issued to a data table. Both these user exits can use screening criteria appropriate to the applications sharing the data.

The MAXNUMRECS parameter provided with the VSAM file definition specifies the maximum number of records that can be in the CFDT itself. The default value for the MAXNUMRECS parameter is NOLIMIT, indicating that there is no limit on the maximum number of records.

A CFDT pool list structure can become full if (1) a large VSAM source data set is loaded into a CFDT, exhausting all structure space; or (2) applications continue to write records to a CFDT, exhausting all structure space.

If the CFDT list structure is allowed to become completely full, message DFHCF0442 (*CF structure strname request failed, structure is full*) is issued and CICS rejects any attempt to add new records to a CFDT or to create new tables in the pool. Additionally, completely filling a CFDT list structure can have a significant impact on performance and application function. IBM gives the following examples of problems that occur when a CFDT pool list structure becomes full:

- Rewrite requests can be rejected even when the size of the new data is less than or equal to the original size.

- Server internal operations can fail, causing internal time-outs and retries.

- An attempt to close a table or change the table status could encounter a structure full condition. In this case, the attempt is retried indefinitely, because it must be completed in order to preserve table integrity (the only alternative would be to terminate the server). The retry process normally succeeds quickly, but there is a theoretical case where this can cause a loop until another server causes temporarily unavailable resources to be released.

- Rewrites with the same (or smaller) data size for a table using the contention update model are retried indefinitely if they initially fail because of a structure full condition. This is done to protect the application against having to handle this unexpected form of failure.

Again, the retry should normally succeed quickly, but there is a theoretical possibility that this could loop for a while.

- Rewrites for a table using the locking or recoverable update model could be rejected with a structure full condition even if the data size is not increased. No retry is attempted in this case.

- Units of work can be backed out because the server is unable to create unit of work control entries for commit processing.

- There may not be sufficient structure space to send lock release messages, in which case waiting tasks are not woken up immediately but continue to wait for up to the time-out interval specified on the LOCKWAITINTERVAL parameter before finding out that the lock has been released.

CFDT pool server startup parameters (ELEMENTRESERVEMIN, ELEMENTRESERVEPC, ENTRYRESERVEMIN and ENTRYRESERVEPC) are provided to reduce the risk of the structure becoming totally full. These parameters reserve a number of entries and elements in the list structure. These can be used only for operations that normally need extra space temporarily, such as rewrites or unit of work control operations.

The number of entries or elements *remaining* in the list structure is returned by each coupling facility access request. If a CFDT pool server is requested to write a new record or create a new table when the space remaining is less than or equal to the specified reserve level, the request is rejected[1] with an indication that no space is available.

Using the reserved space parameters means that, even if the structure fills up very rapidly (for example, because a table is being loaded that is too large for the available space), enough space should remain to allow rewrites of existing records and allow internal communication between servers to continue normally.

This mechanism cannot prevent the structure from eventually becoming totally full, as recoverable rewrites are allowed to use the reserved space temporarily, and rewrites that increase the data length will gradually use up the reserved elements.

CFDT list structure statistics for the coupling facility are available in MXG file CICCFS6D. CPExpert uses data in CICCFS6D to determine whether a Structure Full condition occurred for a CFDT pool server list structure.

[1] Before rejecting the request, the server issues a dummy read request in order to find out the latest usage levels for the structure, in case more space has recently become available.

CICCFS6D variable S6RSP7CT (List structure became full) indicates that the list structure was full.

CPExpert produces Rule CIC425 when the number of Structure Full conditions is greater than the value specified by the **CFNOSPCE** guidance variable in USOURCE(CICGUIDE). The default value for the **CFNOSPCE** is 0, indicating that CPExpert should produce Rule CIC425 whenever any Structure Full conditions occurred.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review the STRUCTURE parameter and INITSIZE parameter in the CFRM policy to determine whether these parameters should be increased.

- Review the *Automatic ALTER parameters* to determine whether these parameters provide sufficient control over the conditions under which the CFDT server attempts an automatic ALTER action when the structure becomes nearly full.

- Review the CFDTs in the pool to see whether any have reached an unreasonably large number of records. Recall that the default value for the MAXNUMRECS parameter is NOLIMIT, indicating that there is no limit on the maximum number of records. A looping application (or one which otherwise generated an unexpectedly large number of records) can flood a CFDT pool with a large number of records. Perhaps a limiting value should be specified for the MAXNUMRECS parameter for any table with an unexpectedly large number of records.

- When a CFDT is loaded, it becomes an independent entity, separate from the behavior of the CICS regions that access the table or caused the table to be loaded. Even when all CICS regions have terminated, either normally or abnormally, a CFDT continues to remain in the coupling facility until explicit action is taken to delete the CFDT's list structure.

  The CFDT contents or list structure can be deleted with a *MODIFY cfdt_server, DELETE TABLE=name* command. Procedures should be reviewed to ensure that any coupling facility data table which is no longer in use (or planned to be in use shortly) is deleted as soon as possible, so that the space can be reused.

- Determine whether the XDTRD user exit has been used for the CFDT. If the XDTRD user exit has **not** been used, consider adding the XDTRD user exit and devising appropriate screening criteria to be applied to the

VSAM KSDS records as they are read by CICS during initial loading[2] of the table. If the XDTRD user exit **has** been used, consider revising the screening criteria in the user exit to exclude a larger number of records during loading.

- Determine whether the XDTAD user exit has been used for the CFDT. If the XDTAD user exit has **not** been used, consider adding the XDTAD user exit and devising appropriate screening criteria to be applied to records as they are written to the CFDT. If the XDTAD user exit **has** been used, consider revising the screening criteria in the user exit to exclude a larger number of records during loading.

- Message DFHCF0446 (*CF structure strname free space is below reserve level. New records will be rejected.*) will be issued if the coupling facility data table server has detected that the number of free list entries or data elements in the pool structure has fallen below the reserve levels specified by the server parameters ENTRYRESERVEMIN, ENTRYRESERVEPC, ELEMENTRESERVEMIN and ELEMENTRESERVEPC.

  Once this message is issued, any request to create a new record or table in the pool will be rejected for as long as the amount of free space remains below the reserve levels. The failing request is given a NOSPACE indication if it originated from a CICS API request.

  Consequently, you should verify that the reserved space parameters (ENTRYRESERVEMIN, ENTRYRESERVEPC, ELEMENTRESERVEMIN and ELEMENTRESERVEPC) have not been specified with values that are too large.

- Change the CFNOSPCE guidance variable in USOURCE(CICGUIDE) so Rule CIC425 is produced only when you wish to be aware of a larger number of Structure Full situations. This alternative is **not** recommended! You should always be aware of Structure Full situations.

- You can "turn off" the rule using the process described in Section 3 of this User Manual. However, this alternative is **not** recommended! You should always be aware of Structure Full situations.

Reference: CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 2.10.8 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 27 (Starting a CFDT server)

[2] Of course, this alternative would not apply if a blank CFDT were created.

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Section 4.3 (Starting a CFDT server)

Rule CIC426: A large percent of list data reads had to be repeated

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of list data reads had to be repeated because the data was larger than the default data transfer size.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The amount of impact depends on how many list data reads had to be repeated and the resulting overhead and delay.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

The default action with this program error is to terminate the task abnormally. However, the application can take remedial action if the HANDLE CONDITION command had been issued before the READQ TS command had been issued.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of list data reads which had to be repeated because the data was larger than the default data transfer size, using the following algorithm:

$$\textit{Percent list data rereads} = \frac{S1RRLCT}{S1RDLCT}$$

where	S1RRLCT	= Number of list data reads that had to be repeated
	S1RDLCT	= Number of list data reads

CPExpert produces Rule CIC426 when the percent of list data reads which had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **CFPCTLDR** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLDR** is 0 indicating that CPExpert should produce Rule CIC426 when any list data reads must be repeated because the data was larger than the default data transfer size.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of shared temporary storage pools to determine whether the READQ TS commands can be issued with a correctly-sized data area for the INTO clause.
- Change the CFPCTLDR guidance variable in USOURCE(CICGUIDE) so Rule CIC426 is produced only when you wish to be aware of applications that cause a larger percent of list data reads must be repeated because the data was larger than the default data transfer size.
- You can specify **%LET CFPCTLDR = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent of list data reads which had to be repeated because the data was larger than the default data transfer size cannot be greater than 100), or you can "turn off" the rule using the process described in Section 3 of this User Manual.

Reference: CICS/TS for OS/390 Release 1.1
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.2
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.3
CICS Application Programming Reference: Section 1.150 (READQ TS)
CICS System Programming Reference: Section 1.5 (Exception conditions)

CICS/TS for z/OS Release 2.1
CICS Application Programming Reference: CICS API Commands (READQ TS)
CICS System Programming Reference: Chapter 1 (Exception conditions)

CICS/TS for z/OS Release 2.2

CICS Application Programming Reference: CICS API Commands (READQ TS)

CICS System Programming Reference: Chapter 1 (Exception conditions)

Rule CIC427: A large percent of index data reads had to be repeated

Finding: The CICS Shared Temporary Storage Queue Server statistics showed that a large percent of index data reads had to be repeated because the data was larger than the default data transfer size.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of the CICS region. The amount of impact depends on how many index data reads had to be repeated and the resulting overhead and delay.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: The READQ TS command reads data from a temporary storage queue. The command optionally specifies INTO(data-area) to describe the data area into which the data is to be written. If the length of the data exceeds the value specified, the data is truncated to that value. A program error (LENGERR) occurs when the length of the stored data is greater than the value specified by the LENGTH option (this condition only applies to the INTO option).

The default action with this program error is to terminate the task abnormally. However, the application can take remedial action if the HANDLE CONDITION command had been issued before the READQ TS command had been issued.

If the request ended prematurely because the buffer was too small to hold the first entry to be read (for instance, the buffer is 4096 bytes but the data entry information is 65536 bytes), the application must determine the size of the data entry for the list entry that caused the failure, and re-issue the READQ TS command with larger buffer areas. This error handling and command re-issue creates unnecessary overhead and delays response.

Shared temporary storage queue server statistics for the coupling facility are available in MXG file CICXQ1. CPExpert uses data in CICXQ1 to calculate the percent of index data reads which had to be repeated because the data was larger than the default data transfer size, using the following algorithm:

$$\textit{Percent index data rereads} = \frac{S1RRQCT}{S1RDQCT}$$

where S1RRQCT = Number of queue index reads that had to be repeated
S1RDQCT = Number of queue index reads

CPExpert produces Rule CIC427 when the percent of queue index reads had to be repeated because the data was larger than the default data transfer size is greater than the value specified by the **CFPCTIDR** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTIDR** is 0, indicating that CPExpert should produce Rule CIC427 when any queue index reads must be repeated because the data was larger than the default data transfer size.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Review application use of shared temporary storage pools to determine whether the READQ TS commands can be issued with a correctly-sized data area for the INTO clause.
- Change the CFPCTIDR guidance variable in USOURCE(CICGUIDE) so Rule CIC427 is produced only when you wish to be aware of applications that cause a larger percent of queue index reads which must be repeated because the data was larger than the default data transfer size.
- You can specify **%LET CFPCTIDR = 100;** in USOURCE(CICGUIDE) to suppress this finding (the percent queue index reads which must be repeated because the data was larger than the default data transfer size cannot be greater than 100), or you can “turn off” the rule using the process described in Section 3 of this User Manual.

Reference: CICS/TS for OS/390 Release 1.1
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.2
CICS Application Programming Reference: Section 1.140 (READQ TS)
CICS System Programming Reference: Section 1.1.5 (Exception conditions)

CICS/TS for OS/390 Release 1.3
CICS Application Programming Reference: Section 1.150 (READQ TS)
CICS System Programming Reference: Section 1.5 (Exception conditions)

CICS/TS for z/OS Release 2.1
CICS Application Programming Reference: CICS API Commands (READQ TS)
CICS System Programming Reference: Chapter 1 (Exception conditions)

CICS/TS for z/OS Release 2.2

CICS Application Programming Reference: CICS API Commands (READQ TS)

CICS System Programming Reference: Chapter 1 (Exception conditions)

Rule CIC440: LOC=ANY request initially failed and was retried

Finding: The CICS Coupling Facility Data Table (CFDT) pool AXM storage statistics showed that a high percent of LOC=ANY storage requests initially failed to obtain the requested storage, and were retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table provides a means of sharing files between CICS regions. Applications access a CFDT using normal file control commands, just as the application would reference records in a shared data table on a single MVS image, but without the need to use a file-owning region (FOR). Consequently, a CFDT eliminates some of the potential capacity problems associated with using an FOR. From an application view, a CFDT appears much like a sysplex-wide user-maintained data table.

CICS automatically creates a CFDT when a first file control command reference requires the CFDT to be opened. This CFDT is then used by the same region, or other CICS regions, that issue subsequent open requests of other files that name the same coupling facility data table.

CICS can either (1) load the coupling facility data table automatically from a source VSAM data set when it is first opened, or (2) the CFDT can be defined with LOAD(NO) specified (which allows creation of an empty CFDT to which records are added).

A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for each CFDT pool defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

When requests in the AXMPGANY pool are retried, this means that all free storage in the AXMPGANY pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the “compress attempt” to reduce fragmentation in the AXMPGANY storage pool will correspondingly increase.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGANY pool initially failed and were retried, using the following algorithm:

$$\textit{Percent AXMPGANY storage requests retried} = \frac{\textit{S9ANYRQC}}{\textit{S9ANYRQG}}$$

where S9ANYRQC = Times a storage request initially failed and was retried
S9ANYRQG = Number of storage GET requests

CPExpert produces Rule CIC440 when the percent of storage requests in the AXMPGANY pool that initially failed and were retried is greater than the value specified by the **CFPCTARC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTARC** is .1, indicating that CPExpert should produce Rule CIC440 whenever more than one tenth percent of the storage requests in the AXMPGANY pool initially failed and were retried.

Suggestion: If this finding is produced, you should consider the following alternatives:

- If this finding occurs often (or if a large percent of requests initially failed and were retried), this could be an indication that storage in the AXMPGANY pool is becoming exhausted and requests might begin failing. Requests are retried only (1) if there are no free areas of the right size and (2) there is not enough storage left in the pool to satisfy the request. Either of these situations occurring frequently could indicate that storage in the pool is in danger of becoming exhausted. If storage is in danger of becoming exhausted, Rule CIC442 (LOC=ANY had low percent minimum free storage) might be produced, but you might have altered the guidance for Rule CIC442 and Rule CIC442 might have been suppressed.

- If storage in the AXMPGANY pool is in danger of becoming exhausted, you should consider increasing the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the **CFPCTARC** guidance variable in USOURCE(CICGUIDE) so Rule CIC440 is produced only when you wish to be aware of a different percent of storage requests in the AXMPGANY pool that initially failed and were retried. Since Rule CIC440 provides an "early warning" of potential impending request failures, you should **not** normally change the CFPCTARC guidance variable.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC441: LOC=ANY requests were unable to obtain storage and failed

Finding: The CICS Coupling Facility Data Table (CFDT) pool storage AXM statistics showed that LOC=ANY storage requests initially failed to obtain the requested storage, were retried after merging adjacent small free areas to form larger areas, and failed because the requested amount of storage was unavailable.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. The coupling facility data table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for <u>each CFDT pool</u> defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC441 when the number of storage requests in the AXMPGANY pool that failed after retry (variable S9ANYRQS[1]) is greater than the value specified by the **CFANYRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFANYRQS** is 0, indicating that CPExpert should produce Rule CIC441 when any storage request in the AXMPGANY pool failed after retry.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be

[1] Please note that the *CICS Performance Guides* describe S9ANYRQF as failed requests and S9ANYRQS as requests to release storage. These descriptions are "reversed" initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9ANYRQF is the requests to "free" or release storage and S9ANYRQS is number of failed requests because of "short on storage".

accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the **CFANYRQS** guidance variable in USOURCE(CICGUIDE) so Rule CIC441 is produced only when you wish to be aware of a different number of storage requests in the AXMPGANY pool that failed after retry. Since Rule CIC441 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFANYRQS guidance variable**.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC442: LOC=ANY had low percent minimum free storage

Finding: CICS Coupling Facility Data Table (CFDT) pool AXM storage statistics showed that there was a low percent of LOC=ANY free storage for the CFDT server pool.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. The coupling facility data table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for <u>each CFDT pool</u> defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICCFS9D to calculate the minimum percent of free storage in the AXMPGANY pool, using the following algorithm:

$$\textit{Minimum percent free AXMPGANY storage} = \frac{\textit{S9ANYLO}}{\textit{S9ANYMX}}$$

where S9ANYLO = Lowest amount of storage that has been free since reset
S9ANYMX = Total number of records in the storage pool

CPExpert produces Rule CIC442 when the percent free storage in the AXMPGANY pool is less than the value specified by the **CFPCTAMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTAMN** is 25, indicating that CPExpert should produce Rule CIC442 whenever less than 25% of storage in the AXMPGANY pool is free.

Suggestion: If this finding is produced, you should consider the following alternatives:

- If storage in the AXMPGANY pool is in danger of becoming exhausted, you should consider increasing the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

  It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the CFPCTAMN guidance variable in USOURCE(CICGUIDE) so Rule CIC442 is produced only when you wish to be aware of a different minimum percent of free storage in the AXMPGANY pool.

.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC443: LOC=BELOW request initially failed and was retried

Finding: The CICS Coupling Facility Data Table (CFDT) AXM storage statistics showed that a high percent of LOC=BELOW storage requests initially failed to obtain the requested storage, and were retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table provides a means of sharing files between CICS regions. Applications access a CFDT using normal file control commands, just as the application would reference records in a shared data table on a single MVS image, but without the need to use a file-owning region (FOR). Consequently, a CFDT eliminates some of the potential capacity problems associated with using an FOR. From an application view, a CFDT appears much like a sysplex-wide user-maintained data table.

CICS automatically creates a CFDT when a first file control command reference requires the CFDT to be opened. This CFDT is then used by the same region, or other CICS regions, that issue subsequent open requests of other files that name the same coupling facility data table.

CICS can either (1) load the coupling facility data table automatically from a source VSAM data set when it is first opened, or (2) the CFDT can be defined with LOAD(NO) specified (which allows creation of an empty CFDT to which records are added).

A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for <u>each CFDT pool</u> defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGLOW storage pool will correspondingly increase.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICCFS9D to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried, using the following algorithm:

$$\textit{Percent AXMPGLOW storage requests retried} = \frac{S9LOWRQC}{S9LOWRQG}$$

where S9LOWRQC= Times a storage request initially failed and was retried
S9LOWRQG = Number of storage GET requests

CPExpert produces Rule CIC443 when the percent of storage requests in the AXMPGLOW pool that initially failed and were retried is greater than the value specified by the **CFPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLRC** is .1, indicating that CPExpert should produce Rule CIC443 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

Suggestion: If this finding is produced, you should consider the following alternatives:

- If this finding occurs often (or if a large percent of requests initially failed and were retried), this could be an indication that storage in the AXMPGLOW pool is becoming exhausted and requests might begin failing. Requests are retried only (1) if there are no free areas of the right size and (2) there is not enough storage left in the pool to satisfy the request. Either of these situations occurring frequently could indicate that storage in the pool is in danger of becoming exhausted. If storage is in danger of becoming exhausted, Rule CIC445 (LOC=BELOW had low percent minimum free storage) might be produced, but you might have altered the guidance for Rule CIC445 and Rule CIC445 might have been suppressed.

- If storage in the AXMPGLOW pool is in danger of becoming exhausted, you should consider increasing the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the **CFPCTLRC** guidance variable in USOURCE(CICGUIDE) so Rule CIC443 is produced only when you wish to be aware of a different percent of storage requests in the AXMPGLOW pool that initially failed and were retried. Since Rule CIC443 provides an "early warning" of potential impending request failures, you should **not** normally change the CFPCTLRC guidance variable.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC444: LOC=LOW requests were unable to obtain storage and failed

Finding: The CICS Coupling Facility Data Table (CFDT) pool storage AXM statistics showed that LOC=LOW storage requests initially failed to obtain the requested storage, were retried after merging adjacent small free areas to form larger areas, and failed because the requested amount of storage was unavailable.

Impact: This finding has a HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. The coupling facility data table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for <u>each CFDT pool</u> defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGLOW* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGLOW pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert produces Rule CIC444 when the number of storage requests in the AXMPGLOW pool that failed after retry (variable S9LOWRQS[1]) is greater than the value specified by the **CFLOWRQS** guidance variable in USOURCE(CICGUIDE). The default value for the **CFLOWRQS** is 0, indicating that CPExpert should produce Rule CIC444 when any storage request in the AXMPGLOW pool failed after retry.

Suggestion: If this finding is produced, you should consider the following alternatives:

- Increase the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be

[1] Please note that the *CICS Performance Guides* describe S9LOWRQF as failed requests and S9LOWRQS as requests to release storage. These descriptions are “reversed” initially, but IBM Hursley corrected the *CICS Performance Guide with document* SC34-6009-05. S9LOWRQF is the requests to “free” or release storage and S9LOWRQS is number of failed requests because of “short on storage”.

accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the **CFLOWRQS** guidance variable in USOURCE(CICGUIDE) so Rule CIC444 is produced only when you wish to be aware of a different number of storage requests in the AXMPGLOW pool that failed after retry. Since Rule CIC444 describes a situation that has a high impact on the performance of the CICS region, **you should not normally change the CFLOWRQS guidance variable**.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC445: LOC=BELOW had low percent minimum free storage

Finding: Coupling Facility Data Table (CFDT) pool AXM storage statistics showed that there was a low percent of LOC=BELOW free storage for the CFDT server pool.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. The coupling facility data table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

A CFDT pool server must be started on each MVS image for each CFDT pool defined in a coupling facility which can be accessed from that MVS image. The Coupling Facility Data Table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

CICS automatically connects to the server for a given CFDT pool the first time that any CFDT within that CFDT pool is referenced. All CFDT pool access is performed by cross-memory calls to the CFDT server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. CPExpert uses data in CICFS9D to calculate the minimum percent of free storage in the AXMPGLOW pool, using the following algorithm:

$$\textit{Minimum percent free AXMPGBELOW storage} = \frac{S9LOWLO}{S9LOWMX}$$

where S9LOWLO = Lowest amount of storage that has been free since reset
S9LOWMX = Total number of pages in the storage pool

CPExpert produces Rule CIC445 when the percent free storage in the AXMPGLOW pool is less than the value specified by the **CFPCTLMN** guidance variable in USOURCE(CICGUIDE). The default value for the **CFPCTLMN** is 25, indicating that CPExpert should produce Rule CIC445 whenever less than 25% of storage in the AXMPGLOW pool is free.

Suggestion: If this finding is produced, you should consider the following alternatives:

- If storage in the AXMPGLOW pool is in danger of becoming exhausted, you should consider increasing the amount of storage that is available for the CFDT pool server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

  In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

  It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTS (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service. Rule CIC446 will be produced if CPExpert's analysis indicates that more CICS regions are using the CFDT pool server than would be expected based on the amount of storage allocated.

- Change the CFPCTLMN guidance variable in USOURCE(CICGUIDE) so Rule CIC445 is produced only when you wish to be aware of a different minimum percent of free storage in the AXMPGLOW pool.

.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

Rule CIC446: CFDT pool server storage allocation was less than expected

Finding: CICS Coupling Facility Data Table (CFDT) AXM storage statistics and the CICS File Control statistics showed that the amount of storage allocated to the CFDT pool server was less than expected.

Impact: This finding has a LOW IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: When Rule CIC440, CIC441, CIC446, CIC443, CIC444, or CIC445 are produced, CPExpert analyzes CICS file assignments to CFDT pools. Rule CIC446 is produced when the amount of storage allocated to the CFDT pool server was less than expected. The finding applies only with CICS/Transaction Server for OS/390 Release 1.3, or CICS/Transaction Server for z/OS.

Discussion: A Coupling Facility Data Table is assigned to a *coupling facility data table pool* in a coupling facility. A CFDT pool consists of an XES list structure on the coupling facility. The coupling facility data table pool can contain one or more CFDTs, and there can be more than one CFDT pool defined for the coupling facility.

Access to a CFDT by CICS transactions running in an AOR is through a *CFDT pool server* that supports a specific named CFDT pool. In this context, the CFDT pool server is similar to a File Owning Region (FOR) that would be used for a normal shared data table.

The CFDT pool server is started in its own region, by executing DFHCFMN. Various parameters are provided to DFHCFMN (POOLNAME, list structure parameters, lock wait parameters, tuning parameters, etc.) to allow tailoring of the data sharing server.

During server initialization, the CFDT server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

In order to prevent overloading the CFDT pool server, the number of CFDT requests that each connected CICS region can have active at a time is limited. The *CICS System Definition Guide* states that this limit is about 10 concurrent requests. Since each request requires about 40KB, the REGION size should specify at least 400KB for each connected CICS region, plus a margin of about 10% for other storage areas. Thus, for a server supporting up to 5 CICS regions, IBM suggests that you should specify REGION=2200K.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

It is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTS (via the CFDT pool server, of course). In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service.

Rule CIC440, CIC441, CIC446, CIC443, CIC444, and CIC445 analyze various potential problems with storage in the AXMPGANY and AXMPGLOW pools. When any of these rules are produced, CPExpert analyzes CICS file assignments to CFDT pools. Rule CIC446 is produced when the amount of storage allocated to the CFDT pool server was less than expected.

The MXG file CICFCR contains statistics related to CICS file control. CPExpert extracts file control information for those files that are assigned to a Coupling Facility Data Table. Included in these statistics is the identification of the CFDT pool server for any data set assigned to a CFDT (MXG variable A17DTCFP contains the name of the CFDT pool to which the data table is assigned).

CPExpert determines the maximum number of CICS regions concurrently using a CFDT pool server. This maximum number is multiplied by ten, to yield the maximum number of requests that could be outstanding for the CFDT pool server. Since each request requires about 40KB, CPExpert multiplies the maximum number of requests by 40KB to yield the maximum storage required for requests. This maximum storage for requests is then multiplied by 110%, to account for other required storage. These calculations yield the maximum amount of storage that should be assigned to the CFDT pool server, considering the number of CICS regions that use the CICS pool server.

Coupling Facility Data Table pool server storage statistics are available in MXG file CICCFS9D. The S9ANYSIZ variable contains the size of the storage pool area for the ANY storage pool statistics, and the S9LOWSIZ variable contains the size of the storage pool area for the LOW storage pool statistics. The sum of these two variables yields the total amount of storage allocated to the CFDT pool on the REGION= parameter.

CPExpert compares the amount of storage that *should be assigned* to the CFDT pool server (considering the number of CICS regions that use the CICS pool server), with the sum of the storage that was allocated to the CFDT pool on the REGION= parameter.

CPExpert produces Rule CIC446 when the amount of storage that *should* be assigned to the CFDT pool server (considering the number of CICS regions that use the CICS pool server) is greater than the sum of the storage that was allocated to the CFDT pool on the REGION= parameter.

Suggestion: If this finding is produced, you should consider the following alternatives:

- You should increase the storage allocated to the CFDT pool server (using the REGION= parameter). As mentioned earlier, it is possible that the REGION size was specified correctly initially, but additional CICS regions began using CFDTs. In this case, it is possible that the REGION size for the CFDT pool server was not updated to account for the additional CICS regions requiring service.

- You can "turn off" Rule CIC446 if you do not wish to be informed of the data produced by this finding. Section 3 of the CICS Component User Manual describes how to "turn off" rules.

.

Reference: *CICS/TS for OS/390 Release 1.3*

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.6.13 (Coupling facility data tables)

CICS/TS for z/OS Release 2.1

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)

CICS/TS for z/OS Release 2.2

CICS System Definition Guide: Section 4.3.2 (Defining and starting a coupling facility data table server region)

CICS Performance Guide: Section 4.5.13 (Using coupling facility data tables to gain performance benefits)