Section 2: Specifying Guidance Variables

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

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

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

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

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

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

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

Chapter 1: Data Selection and Presentation Variables

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

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

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

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

EXHIBIT 2-1

Chapter 1.1: Specifying data selection guidance

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

Chapter 1.1.1: WLMDATES and WLMTIMES variables

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

```
%LET WLMDATES = 08MAR1995;      * START DATE FOR DATA ANALYSIS;
%LET WLMTIMES = 08:00:00;       * START TIME FOR DATA ANALYSIS;
```

Chapter 1.1.2: WLMDATEE and WLMTIMEE variables

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

```
%LET WLMDATEE = 08MAR1995;      * END DATE FOR DATA ANALYSIS;
%LET WLMTIMEE = 17:00:00;       * END TIME FOR DATA ANALYSIS;
```

Chapter 1.1.3: WLMDAT2S and WLMTIM2S variables

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

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

```
%LET WLMDAT2S = 08MAR1995;      * START DATE FOR DATA ANALYSIS;
%LET WLMTIM2S = 20:00:00;       * START TIME FOR DATA ANALYSIS;
```

Chapter 1.1.4: WLMDAT2E and WLMTIM2E variables

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

```
%LET WLMDAT2E = 08MAR1995;      * END DATE FOR DATA ANALYSIS;
%LET WLMTIM2E = 22:00:00;       * END TIME FOR DATA ANALYSIS;
```

Chapter 1.1.5: SHIFT variable

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

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

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

Chapter 1.1.6: N_PLEX variable

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

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

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

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

Chapter 1.1.7: SYSPLEX variable

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

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

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

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

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

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

Chapter 1.1.8: SYSTEM variable

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

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

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

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

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

Chapter 1.1.9: SYSTEMn variables

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

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

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

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

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

Chapter 1.1.10: SELECTSW, SELECTn, and PERIODn variables

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

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

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

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

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

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

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

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

SELECTSW variable

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

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

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

SELECTn variables

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

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

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

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

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

PERIODn variables

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

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

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

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

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


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

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

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

Chapter 1.1.11: EXCLUDSW and EXCLUDEn variables

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

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

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

EXCLUDSW variable

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

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

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

EXCLUDEn variable(s)

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

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

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

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

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

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

Chapter 1.2: Specifying data presentation guidance

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

Chapter 1.2.1: VERBOSE variable.

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

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

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

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

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

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

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

Chapter 1.2.2: LISTRPT variable

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

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

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

Chapter 1.2.3: POLORDER variable

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

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

Chapter 1.2.4: WLMAST variable

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

Chapter 1.3: SAS Output Delivery System

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

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

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

Chapter 2: Analysis Guidance Variables

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

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

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

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

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

EXHIBIT 2-2

Chapter 2.1: AVGRESP variable

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

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

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

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

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

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

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

Chapter 2.2: CHKPLCY variable

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

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

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

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

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

Chapter 2.3: EXECSAMP variable

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

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

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


Chapter 2.4: HIGHCPU variable

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

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

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

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

Chapter 2.5: INACTIVE variable

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

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

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

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


Chapter 2.6: MAXDUR variable

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

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

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

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

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

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

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

Chapter 2.7: MAXRESP variable

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

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

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

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

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

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

Chapter 2.8: MAXVEL variable

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

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

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

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

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

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

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

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

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

Chapter 2.9: MINSAMP variable

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

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

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


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

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

Chapter 2.10: MINTRANS variable

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

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

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

Chapter 2.11: OKPAGEIN variable

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

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

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

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

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

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

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

Chapter 2.12: PCTSERVC variable

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

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


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

Chapter 2.13: PERFINDX variable

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

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

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

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

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

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

Chapter 2.14: PHASE variable

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

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

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

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

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

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

Chapter 2.15: POLCHG variable

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

---

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

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

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

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

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

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

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

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

Chapter 2.16: SWAPDW variable

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

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

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

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

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

Chapter 2.17: SWAPTO variable

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

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

RULE WLM070 will be produced if the below conditions apply:

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

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

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

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

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


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

Chapter 2.18: Turning OFF WLM Component Rules

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

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

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

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

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

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

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

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

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

Chapter 2.19: WLMSIG variable

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

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

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

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

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


Chapter 3: XCF Performance Analysis Guidance Variables

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

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

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

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

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

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

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

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

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

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

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

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

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

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


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

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

DEFAULT VALUES FOR XCF ANALYSIS IN USOURCE(WLMGUIDE)

EXHIBIT 2-3

Chapter 3.1: COUPLE variable

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

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

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

Chapter 3.2: EXCLASSW and EXCLASSn variables

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

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

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

Chapter 3.2.1: EXCLASSW variable

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

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

Chapter 3.2.2: EXCLASSn variable(s)

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

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

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

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

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

Chapter 3.3: PCTSML variable

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

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

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

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

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

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

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

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

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

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

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

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

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


Chapter 3.4: PCTBIG variable

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

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

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

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

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

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

Chapter 3.5: PCTREJ variable

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

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

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

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

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

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

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

Chapter 3.6: LOCKCONT variable

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

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

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

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

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

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

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

Chapter 3.7: FALSECNT variable

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

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

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

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

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

Chapter 3.8: SYNCSRV variable and LOCKSRV

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

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

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

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

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

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

Chapter 3.9: ASYNCSRV variable

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

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

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

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

Chapter 3.10: SYNCCHG variable

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

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

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

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

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

Chapter 3.11: Specifying guidance for specific structures

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

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

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

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

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

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

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

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

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

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

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

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


Chapter 4: System Logger Analysis Guidance Variables

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

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

Applications that use the system logger services include:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

DEFAULT VALUES FOR SYSTEM LOGGER ANALYSIS

EXHIBIT 2-4

Chapter 4.1: SMFTYP88 variable

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

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

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

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

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

Chapter 4.2: LGDSFULL

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

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

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

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

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

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

Chapter 4.3: LGSHIFTS

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

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

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

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

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

Chapter 4.4: PCTINTST

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

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

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

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

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

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

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

where:

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

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

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

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

Chapter 4.5: PCTLOCT

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

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

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

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

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

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

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

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

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

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

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

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

where

SMF88SAB = Bytes deleted after being offloaded

SMF88SIB = Bytes deleted before being offloaded

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

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

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

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

Chapter 4.6: STDSHIGH

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Chapter 4.7: STFULL90

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

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

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

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

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

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

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

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

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

Chapter 4.8: STRC2

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

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

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

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

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

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

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

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

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

Chapter 4.9: STRC3

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

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

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

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

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

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

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

Chapter 4.10: STRFULL

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

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

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

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

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

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

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

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

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

Chapter 4.11: Specifying guidance for specific log streams

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

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

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

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

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

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

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

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

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

